HIGH PERFORMANCE MODELING OF CIRCULATION AND
TRANSPORT IN THE INDIAN RIVER LAGOON, FLORIDA

By

JUSTIN ROSS DAVIS

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2001

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

HIGH PERFORMANCE MODELING OF CIRCULATION AND
TRANSPORT IN THE INDIAN RIVER LAGOON, FLORIDA

By

Justin Ross Davis

December 2001

Chairman: Dr. Y. Peter Sheng
Major Department: Civil and Coastal Engineering

This study explores the use of shared memory and message passing parallel techniques to the simulation of circulation and transport in the Indian River Lagoon, Florida. A parallel version of the CH3D model was developed using shared memory techniques, tested and applied to year-long simulations of circulation and salinity transport in the lagoon. The original CH3D source code was modified using specially designed parallel macros. Speedup and efficiency for the hydrodynamic, salinity, flushing, sediment and water quality models were determined under three computing environments: a four processor SGI Origin 2000, a two processor SGI Octane and a two processor Dell PowerEdge 1300 running Linux. For example, a speedup of 3.01 was obtained using four processors of the Origin 2000. Using the developed parallel CH3D model, a series of one-year hydrodynamic and salinity transport simulations were performed. The model was able to simulate mean water level to within 3.44% of the measured values and RMS error between the simulated and measured water level, and the simulated and measured salinity was 5.31% and 18.07%, respectively. The

parallel CH3D model was shown to be conservative and produced the same results as the serial CH3D model.

An efficient parallel, two-dimensional, vertically averaged, wetting and drying circulation model was developed using message passing techniques. The model solved for the Coriolis terms implicitly and, in conjunction with a simple storm wind and pressure model, can be used for prediction of storm surges due to hurricanes. The model was validated using several analytic test cases and used to successfully simulate the storm surge in the Indian River Lagoon during the passage of Hurricanes Floyd and Irene (1999). The model was then used to simulate various hypothetical hurricanes making landfall in the lagoon. The maximum flooding was predicted to occur when the storm made landfall two hours after high tide and which can be greater than 7 m when a major Category 5 hurricane made landfall.

CHAPTER 1
INTRODUCTION

The health of our nation's water bodies has become a major issue in today's society. Federal and local agencies are spending countless dollars to restore the health and prevent further degradation of numerous polluted lakes, estuaries, and coastal waters. Numerous laboratory, field and numerical modeling studies are being performed by scientists and engineers in an effort to better understand and quantify the complex physical, chemical and biological processes present. Laboratory studies have been proven to be useful for quantitative understanding of small-scale and local processes while field studies are more effective in quantifying large-scale and global processes. Finally, through a process of model building, calibration, and validation, it is possible to develop numerical models to simulate the dynamics of various aquatic ecosystems.

Numerical models can be used not only to study the past and the present but also to predict the future. Numerical models can be used to simulate the response of an aquatic ecosystem to a variety of hypothetical anthropogenic and natural changes. Example hypothetical changes include increasing or decreasing the inflow of nutrients into an estuary or the flooding associated with major tropical storms.

1.1 Field and Modeling Studies at UF

At the Civil and Coastal Engineering Department of the University of Florida (UF), since 1986, Dr. Y. Peter Sheng's group has conducted many field and modeling studies to quantitatively understand the hydrodynamics, water quality, and ecological processes in lakes, estuaries, and coastal waters. These studies have led to

development of hydrodynamic models for the Chesapeake Bay (Sheng, 1990), James River (Sheng et al., 1990b; Choi, 1992), Lake Okeechobee (Sheng et al., 1990a,c; Sheng and Lee, 1991; Lee and Sheng, 1993; Chen, 1994), Sarasota Bay and Tampa Bay (Sheng and Peene, 1994; Peene and Sheng, 1995), Florida Bay (Sheng, 1995; Davis, 1996), Tampa Bay (Sheng and Yassuda, 1994), Lake Apopka (Sheng and Meng, 1993) and Indian River Lagoon (Sheng et al., 1990d, 1994; Sheng, 1997, 2000). Water quality models which include the modeling of nitrogen cycling and phosphorus cycling have been developed for Lake Okeechobee (Sheng et al., 1993a; Chen and Sheng, 1996), Roberts Bay (Sheng et al., 1995, 1996b) and Tampa Bay (Sheng et al., 1993b; Yassuda, 1996). The studies of Roberts Bay (Sheng et al., 1995) and Tampa Bay (Yassuda, 1996) include the modeling of seagrass in addition to hydrodynamics and water quality.

During the past two decades, these studies have advanced from using simple 1-D cross-sectionally averaged models (*e.g.*, Sheng et al. (1990d)) to using fully 3-d models complete with coupled sediment, wave, nutrient, light and seagrass models to further the understanding of estuaries as integrated systems (Sheng, 1997, 2000). Along with the inclusion of more complicated processes into the models, there have been significant increases in simulation domain, grid resolution and simulation period (Table 1.1). Unfortunately, as the complexity of these integrated models increases, their execution time substantially increases as well.

1.2   The IRL-PLR Study

In 1994, major research work was begun to develop an Indian River Lagoon Pollutant Load Reduction (IRL-PLR) model for the St. Johns River Water Management District (SJRWMD) (Sheng, 1997, 2000). The IRL lies on the Atlantic coast of central Florida (Figure 1.1) and is one of the most biologically diverse and complex ecosystems in the United States, in terms of habitat and species (Barile et al., 1987). The lagoon is approximately 240 km long and extends from Ponce de Leon Inlet in

Table 1.1: Evolution of grid sizes used in modeling studies in the Civil and Coastal Engineering Department during the past two decades.

Year	Water Body	Grid Size	Total Grid Cells	Simulation Period (days)
1983	Mississippi Sound (Sheng, 1983)	60x116x4	28,000	5
1988	Lake Okeechobee (Liu, 1988)	31x50x1	2,000	30
1991	Sarasota/Tampa Bay (Peene et al., 1993)	189x50x4	38,000	60
1993	Yellow Sea (Luo, 1993)	50x69x5	17,000	30
1996	Florida Bay (Davis, 1996)	194x148x1	29,000	60
1996	Indian River Lagoon (Sheng et al., 1996a)	478x44x4	84,000	60
1997	Tampa Bay (Yassuda and Sheng, 1998)	45x85x8	32,000	120
1999	Charlotte Harbor	63x105x8	53,000	90
1999	Pinellas County	104x185x4	77,000	90
2002	Indian River Lagoon-3x(proposed)	1434x132x4	757,000	365
2004	Indian River Lagoon-5x(proposed)	2390x220x8	4,206,000	365

the north to Jupiter Inlet in the south. The lagoon is 2-4 km wide with an average depth of 2 m outside of the Intracoastal Waterway (ICW). The lagoon receives fresh water from numerous natural creeks, rivers and man-made canals.

The IRL-PLR model is based on the CH3D model (Sheng, 1983) and includes hydrodynamic, sediment transport, water quality, light attenuation and seagrass models. The study also includes the collection and analysis of extensive hydrodynamic, sediment, water quality, light and seagrass data (Sheng et al., 1998a,b, 1999; Sheng and Davis, 1999). Part of the study presented herein is the development, application and verification of the hydrodynamic model used in the IRL-PLR study.

As part of the IRL-PLR study, numerous year-long simulations of hydrodynamics, sediment, and water quality in the IRL will need to be performed. Preliminary coupled hydrodynamic, sediment, water quality and seagrass models simulations indicated long execution times for the integrated model. During the development of the CH3D model used in the IRL-PLR study, numerous code optimizations were made in an attempt to significantly decrease execution time of the model; however, none of these changes proved effective. In an attempt to reduce the execution time

Figure 1.1: The Indian River Lagoon (Davis and Sheng, 2000).

Shared communication link
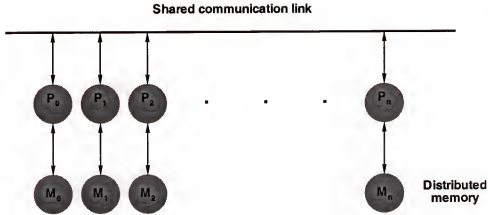
Figure 1.2: A schematic representation of the message passing model (Davis and Sheng, 2000).

of the IRL-PLR model, another part of the study presented herein is the parallel implementation of the CH3D model.

1.3 Parallel Communication Models

Parallel computing can be divided into two predominant communication models, the shared memory model (Figure 1.3) and the message passing model (Figure 1.2). The shared memory (also known as multi-thread) model relies on a globally accessible area of shared memory which can be accessed by the individual processing elements (CPUs). The message passing model requires that data explicitly be sent from one processing element and explicitly received by another. The shared memory model, while easier to implement, does not scale well with the number of processors. The message passing model, on the other hand, scales better because the message passing is explicit and the application programmer can provide for the most efficient communication.

The shared memory model is generally implemented with parallel constructs which are added to the serial source code. These constructs are slightly different between compiler vendors but a standard called OpenMP[1] is now being supported

---

[1] http://www.openmp.org
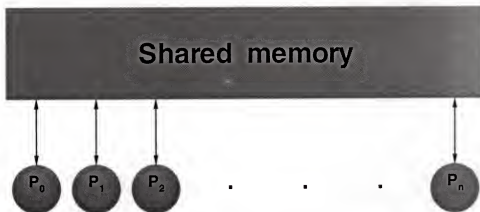
Figure 1.3: A schematic representation of the shared memory model (Davis and Sheng, 2000).

by major compiler vendors (SGI, HP, Intel, Sun, Compaq, Absoft, Portland Group, *etc.*).

The message passing model is generally implemented with explicit library calls to send and receive data. Several different libraries exist to facilitate the passing of data, *e.g.*, MPI[2] (Message Passing Interface Forum, 1994) (*e.g.* LAM[3] and MPICH[4] (Gropp and Lusk, 2001)), PVM[5] (Beguelin et al., 1991), and shmem (SGI proprietary).

The CH3D model was developed for serial computation and conversion to parallelism via a message passing model would be extremely difficult. In spite of the poor processor scaling, shared memory techniques were used to parallelize the CH3D model. However, to explore the applicability of parallelism via the message passing model, a new model was developed, verified, and applied herein. Applications of the newly developed message passing model are designed to study the storm surge associated with the passage and impact of hurricanes on the IRL.

---

[2]http://www-unix.mcs.anl.gov/mpi/

[3]http://www.lam-mpi.org/

[4]http://www-unix.mcs.anl.gov/mpi/mpich/

[5]http://www.epm.ornl.gov/pvm/pvm_home.html

<u>1.4 Hurricanes' Effect on Florida</u>

Hurricanes are one of the most dangerous and damaging natural phenomena. As a hurricane approaches and impacts the coastline, storm surge flooding, strong winds, torrential rainfall and tornados can reek havoc on both coastal and inland communities. However, of all these impacts, storm surge accounts for over 90% of hurricane related deaths (Pielke, 1990). The intensity of a hurricane is generally referred to using the Saffir-Simpson scale (Table 1.2). The scale ranges from relatively weak, a "Category 1" hurricane, to completely devastating, a "Category 5" hurricane.

Table 1.2: The Saffir-Simpson hurricane intensity scale (Simpson and Riehl, 1981).

Category	Wind Speed	Barometric Pressure	Storm Surge	Damage
1	75 - 95 mph	$\geq$ 28.94 in hg	4.0 - 5.0 ft	Minimal
(Weak)	33.5 - 42.5 m/s	$\geq$ 980.02 mb	1.2 - 1.5 m	
2	96 - 110 mph	28.93 - 28.50 in hg	6.0 - 8.0 ft	Moderate
(Moderate)	42.9 - 49.2 m/s	979.68 - 965.12 mb	1.8 - 2.4 m	
3	111 - 130 mph	28.49 - 27.91 in hg	9.0 - 12.0 ft	Extensive
(Strong)	49.6 - 58.1 m/s	964.78 - 945.14 mb	2.7 - 3.7 m	
4	131 - 155 mph	27.90 - 27.17 in hg	13.0 - 18.0 ft	Extreme
(Very Strong)	58.6 - 69.3 m/s	944.80 - 920.08 mb	3.9 - 5.5 m	
5	> 155 mph	< 27.17 in hg	> 18 ft	Catastrophic
(Devastating)	> 69.3 m/s	< 920.08 mb	> 5.5 m	

The Florida peninsula extends out into the warm waters of the Gulf of Mexico and Atlantic Ocean where hurricanes form thus making Florida highly susceptible to landfalling hurricanes. Every couple of years since 1900 a hurricane has made landfall in Florida (Figure 1.4). Because of the high occurrence of landfalling hurricanes in Florida, three of the top five most intense (Table 1.3), most costly (Table 1.4) and most deadly (Table 1.5) hurricanes have impacted Florida. A list of the destruction of some of Florida's most notorious storms is shown in Table 1.6.

<u>1.5 Objectives of this Study</u>

The objectives of this study are the following:
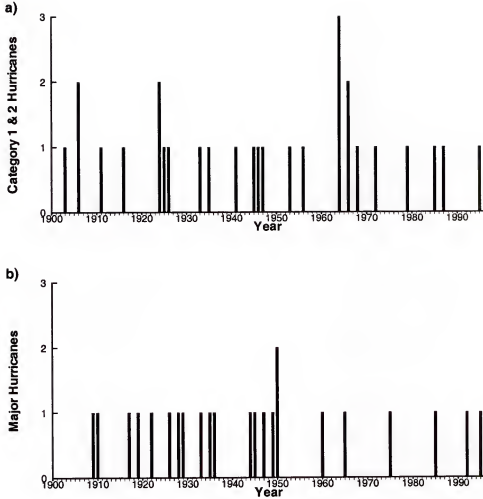
Figure 1.4: Number of a) minor (Category 1 and 2) and b) major (Category 3, 4 and 5) hurricanes per year which make landfall in Florida (Landsea, 2000).

- Review the governing equations of the CH3D model and develop a grid system for study of the IRL (Chapter 2).

- Explore parallelization methods for the CH3D model, develop a parallel version of the existing CH3D hydrodynamic model and create an associated parallel grid system for study of the IRL (Chapter 3).

- Provide an overview of measured data available within the study area; use the newly developed parallel CH3D model to perform one year simulations of the IRL; compare simulated results, both qualitatively and quantitatively, with

Table 1.3: The ten most intense hurricanes at time of landfall from 1900 to 1999 (Landsea, 2000).

Rank	Hurricane	Year	Category	Pressure (in hg)
1	FL (Keys)	1935	5	26.35
2	Camille (MS/SE LA/VA)	1969	5	26.84
3	Andrew (SE FL/SE LA)	1992	4	27.23
4	FL (Keys)/S TX	1919	4	27.37
5	FL (Lake Okeechobee)	1928	4	27.43
6	Donna (FL/Eastern U.S.)	1960	4	27.46
7	TX (Galveston)	1900	4	27.49
7	LA (Grand Isle)	1909	4	27.49
7	LA (New Orleans)	1915	4	27.49
7	Carla (N & Cent. TX)	1961	4	27.49

Table 1.4: The ten most costly hurricanes from 1900 to 1999 (Landsea, 2000). Damages are given in 1998 U.S. dollars.

Rank	Hurricane	Year	Category	Damages
1	SE Florida/Alabama	1926	4	$83.8 billion
2	Andrew (SE FL/LA)	1992	4	$38.4 billion
3	N Texas (Galveston)	1900	4	$30.9 billion
4	N Texas (Galveston)	1915	4	$26.1 billion
5	SW Florida	1944	3	$19.5 billion
6	New England	1938	3	$19.3 billion
7	SE Florida/Lake Okeechobee	1928	4	$16.0 billion
8	Betsy (SE FL/LA)	1965	3	$14.4 billion
9	Donna (FL/Eastern U.S.)	1960	4	$14.0 billion
10	Camille (MS/LA/VA)	1969	5	$12.7 billion

measured data; perform a timing analysis of the parallel CH3D model; and verify that both serial and parallel models produce identical results (Chapter 4).

- Develop a new parallel hydrodynamic model to simulate hurricane induced storm surges (Chapter 5).

Table 1.5: The ten most deadly hurricanes from 1900-1999 (Landsea, 2000).

Rank	Hurricane	Year	Category	Deaths
1	Unnamed - Galveston, TX	1900	4	8000+
2	Unnamed - Lake Okeechobee, FL	1928	4	1836
3	Unnamed - Fl Keys/S TX	1919	4	600
4	"New England"	1938	3	600
5	"Labor Day" - FL Keys	1935	5	408
6	Audrey - SW LA/N TX	1957	4	390
7	Unnamed - NE U.S.	1944	3	390
8	Unnamed - Grand Isle, LA	1909	4	350
9	Unnamed - New Orleans, LA	1915	4	275
10	Unnamed - Galveston, TX	1915	4	275

Table 1.6: A selection of notorious Florida Hurricanes from 1900 to 1997 (Barnes, 1998). Maximum winds are a) estimated gusts, b) measured one-minute velocities or c) measured gusts. Florida damages are given in unadjusted U.S. dollars.

Hurricane	Category	Maximum Wind (mph)	Pressure (in hg)	Florida Deaths	Florida Damages
September 1906	2	100[a]		60+	$2+ million
October 1906	2	100+[a]	28.55	193+	$160,000
September 1919	4		27.37	300	$2 million
October 1921	3		28.11	8+	$3 million
September 1926	4	150[a]	27.61	243	$112 million
September 1928	4	150[a]	27.43	1,836	$26 million
September 1929	3	150[a]	27.99	3	$821,000
September 1933	3	125[a]	27.99	2	$4 million
September 1935	5	200[a]	26.35	408	$11 million
September 1945	3	196[a]	28.08	4	$60 million
September 1947	4	155[b]	27.76	17	$32 million
August 1949	3	153[b]	28.17	2	$45 million
King, 1950	3	150[c]	28.20	3	$24+ million
Donna, 1960	4	175[a]	27.46	13	$300+ million
Betsy, 1965	3	165[a]	28.12	13	$139 million
Eloise, 1975	3	135[c]	28.20	2	$150 million
Andrew, 1992	4	177[a]	27.23	43	$25+ billion
Opal, 1995	3	144[c]	27.82	2	$3 billion

- Test and validate the new model's accuracy using several analytical tests (Chapter 6).

- Develop several parallel IRL grid systems for use with the newly developed storm surge model and then simulate the passage of several tropical storms (Chapter 7).

- Finally, present a summary of the work performed and conclusions of this study (Chapter 8).

CHAPTER 2
THE CH3D MODEL AND IRL GRID SYSTEM

CH3D, a three-dimensional curvilinear-grid hydrodynamic model, was originally developed by Sheng (1987, 1990, 1994). The CH3D model first solves the vertically-integrated equations of motion, before solving the equations for the deficit horizontal velocities (the difference between the vertically-varying horizontal velocities and the vertically-integrated velocities) and the vertical velocities. In this chapter, the governing equations of the CH3D model are reviewed and the boundary-fitted grid systems used for IRL simulations are developed. More information on the governing equations can be found in Sheng (1987), Sheng (1994) and Sheng et al. (1990b).

2.1 Governing Differential Equations

The governing three-dimensional Cartesian equations describing free surface flows can be derived from the Navier-Stokes equations. After Reynold's averaging, and applying the hydrostatic and Boussinesq approximations, the continuity equation and x- and y-momentum equations have the following form (Sheng, 1983):

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \tag{2.1}$$

$$\frac{\partial u}{\partial t} + \frac{\partial uu}{\partial x} + \frac{\partial uv}{\partial y} + \frac{\partial uw}{\partial z} = -g\frac{\partial \zeta}{\partial x} + fv + A_H\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) + \frac{\partial}{\partial z}\left(A_v\frac{\partial u}{\partial z}\right) \tag{2.2}$$

$$\frac{\partial v}{\partial t} + \frac{\partial vu}{\partial x} + \frac{\partial vv}{\partial y} + \frac{\partial vw}{\partial z} = -g\frac{\partial \zeta}{\partial y} - fu + A_H\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) + \frac{\partial}{\partial z}\left(A_v\frac{\partial v}{\partial z}\right) \tag{2.3}$$

where $u(x, y, z, t)$, $v(x, y, z, t)$, $w(x, y, z, t)$ are the velocity components in the horizontal, x- and y-, and the vertical, z-, directions; t is the time; $\zeta(x, y, t)$ is the free surface

12

elevation; g is the gravitational acceleration; and A_H and A_v are the horizontal and vertical turbulent eddy coefficients, respectively.

In Cartesian coordinates, the conservation of salt and temperature can be written as

$$\frac{\partial S}{\partial t} + \frac{\partial uS}{\partial x} + \frac{\partial vS}{\partial y} + \frac{\partial wS}{\partial z} = \frac{\partial}{\partial x}\left(D_H \frac{\partial S}{\partial x}\right) + \frac{\partial}{\partial y}\left(D_H \frac{\partial S}{\partial y}\right) + \frac{\partial}{\partial z}\left(D_v \frac{\partial S}{\partial z}\right) \quad (2.4)$$

$$\frac{\partial T}{\partial t} + \frac{\partial uT}{\partial x} + \frac{\partial vT}{\partial y} + \frac{\partial wT}{\partial z} = \frac{\partial}{\partial x}\left(K_H \frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(K_H \frac{\partial T}{\partial y}\right) + \frac{\partial}{\partial z}\left(K_v \frac{\partial T}{\partial z}\right) \quad (2.5)$$

where S is salinity, T is temperature, D_H and K_H are the horizontal turbulent eddy diffusivity coefficients for salinity and temperature, respectively, and D_V and K_V are the vertical turbulent eddy diffusivity coefficients for salinity and temperature, respectively.

Various forms of the equation of state can be used. The present model uses the equation given by (Eckart, 1958)

$$
\begin{aligned}
\rho =\ & \frac{P}{\alpha + 0.698P} \\
P =\ & 5890 + 38T - 0.375T^2 + 3S \\
\alpha =\ & 1779.5 + 11.25T - 0.0745T^2 - (3.8 + 0.01T)S
\end{aligned}
\quad (2.6)
$$

where T is in degrees Celsius, S is in ppt and p is in $\frac{g}{cm^3}$.

The governing equations are non-dimensionalized using the following reference scales: X_r and Z_r are the reference lengths in the vertical and horizontal directions; U_r is the reference velocity; ρ_r, ρ_o and $\Delta\rho = \rho - \rho_o$ are the reference density, mean density and density gradient in a stratified flow; A_{Hr} and A_{Vr} are the reference eddy viscosities in the horizontal and vertical directions; D_{Hr} and D_{Vr} are the reference eddy diffusivities in the horizontal and vertical directions. The dimensionless variables can then be written as (Sheng, 1983)

$$(x^*, y^*) = \frac{(x,y)}{X_r}$$
$$z^* = \frac{z}{Z_r}$$
$$(u^*, v^*) = \frac{(u,v)}{U_r}$$
$$w^* = \frac{w \frac{X_r}{Z_r}}{U_r}$$
$$\omega^* = \omega \frac{X_r}{U_r}$$
$$t^* = t f$$
$$(\tau_x^*, \tau_y^*) = \frac{(\tau_x^w, \tau_y^w)}{\rho_o f Z_r U_r} = \frac{(\tau_x^w, \tau_y^w)}{\tau_r}$$
$$\zeta^* = \frac{g\zeta}{f U_r Z_r} = \frac{\zeta}{\zeta_r}$$
$$A_H^* = \frac{A_H}{A_{Hr}}$$
$$A_v^* = \frac{A_v}{A_{vr}}$$
$$D_H^* = \frac{D_H}{D_{Hr}}$$
$$D_v^* = \frac{D_v}{D_{vr}}$$

$$(2.7)$$

These dimensionless variables can be combined yielding the following dimensionless parameters

$$
\begin{array}{ll}
\text{Rossby Number :} & R_o = \frac{U_r}{f X_r} \\
\text{Froude Number :} & F_r = \frac{U_r}{\sqrt{g Z_r}} \\
\text{Densimetric Froude Number :} & F_{rd} = \frac{F_r}{\sqrt{\epsilon}} \\
\text{Vertical Ekman Number :} & E_v = \frac{A_{vr}}{f Z_r^2} \\
\text{Horizontal Ekman Number :} & E_H = \frac{A_{Hr}}{f X_r^2} \\
\text{Vertical Schmidt Number :} & S_{c_v} = \frac{A_{vr}}{D_{vr}} \\
\text{Horizontal Schmidt Number :} & S_{c_H} = \frac{A_{Hr}}{D_{Hr}} \\
& \epsilon = \frac{\rho_r - \rho_o}{\rho_o} \\
& \beta = \frac{g Z_r}{f^2 X_r^2} = \left(\frac{R_o}{F_r} \right)^2
\end{array}
$$

$$(2.8)$$

In the presence of a complex shoreline, a "boundary-fitted" grid allows accurate representation of lateral boundaries. Using the elliptic grid generation technique developed by Thompson (1982) and Thompson et al. (1985), a non-orthogonal boundary-fitted grid can be generated in the horizontal directions. To solve for flow in a boundary fitted grid, it is necessary to transform the governing equations from the original coordinates (x, y) to the transformed coordinates (ξ, η). Computationally, the boundary-fitted grid is transformed from a prototype to a transformed system (Figure 2.1). During the transformations, the velocities are transformed into contravariant velocities. These transformations and further details can be found in the works of Sheng (1986, 1987, 1990).
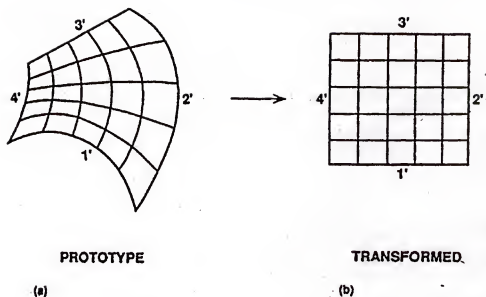
Figure 2.1: A boundary-fitted grid in the a) prototype and b) transformed systems.

In three-dimensional modeling, complex bottom topographies can be better represented with the application of σ-stretching (Sheng, 1983). This transformation allows the same vertical resolution in the shallow coastal areas as well as the deeper navigation channels. The vertical coordinate, z, is transformed into a new coordinate, σ, by (Phillips, 1957)

$$\sigma = \frac{z - \zeta(x, y, t)}{h(x, y) + \zeta(x, y)} \tag{2.9}$$

where h is the water depth measured to a vertical datum (Figure 2.2). Whereas $w = \frac{dz}{dt}$ in the z-plane, $\omega = \frac{d\sigma}{dt}$ in the σ-plane. Using this new vertical coordinate system, the vertical velocity is calculated using

$$w = H\omega + (1 + \sigma)\frac{D\zeta}{Dt} + \sigma\left(u\frac{dh}{dx} + v\frac{dh}{dy}\right) \tag{2.10}$$

In the boundary-fitted, curvilinear, non-dimensional coordinate system, the continuity and u and v-momentum equations are

$$\frac{\partial\zeta}{\partial t} + \frac{\beta}{\sqrt{g_o}}\left[\frac{\partial}{\partial\xi}(\sqrt{g_o}Hu) + \frac{\partial}{\partial\eta}(\sqrt{g_o}Hv)\right] + \beta\frac{\partial H\omega}{\partial\sigma} = 0 \tag{2.11}$$

whose inverse is

$$g^{ij} = \frac{1}{J^2} \begin{bmatrix} x_\eta{}^2 + y_\eta{}^2 & -(x_\xi x_\eta + y_\xi y_\eta) \\ -(x_\eta x_\xi + y_\eta y_\xi) & x_\xi{}^2 + y_\xi{}^2 \end{bmatrix} = \begin{bmatrix} g^{11} & g^{12} \\ g^{21} & g^{22} \end{bmatrix} \quad (2.16)$$

As shown in Sheng (1986), the contravariant components (u^i) and physical components $(u(i))$ of the velocity vector in the non-Cartesian system are locally parallel or orthogonal to the grid lines, while the covariant components (u_i) are generally not parallel or orthogonal to the local grid lines. The three components are identical in a Cartesian coordinate system. The relationship between the physical velocity and the contravariant and covariant velocities are given by

$$u^i = \frac{u(i)}{\sqrt{g_{ii}}} \quad (2.17)$$

$$u_i = \frac{g_{ij}}{\sqrt{g_{ii}}} u(j) \quad (2.18)$$

with no summation on i. The relationship between the contravariant velocities and covariant velocities in the prototype and transformed planes are given by

$$\bar{u}^i = \frac{\partial \xi^i}{\partial x^j} u^j \quad (2.19)$$

$$\bar{u}_i = \frac{\partial \xi^j}{\partial x^i} u_j \quad (2.20)$$

where the unbarred quantities represent the components in the prototype system and the barred quantities represent the components in the transformed system.

The salinity transport equation can be written as

$$\begin{aligned}
\frac{\partial HS}{\partial t} = & \; \frac{E_v}{HS_{c_v}} \frac{\partial}{\partial \sigma} \left(D_v \frac{\partial S}{\partial \sigma} \right) - R_o \frac{\partial H\omega S}{\partial \sigma} \\
& - \frac{R_o}{\sqrt{g_o}} \left[\frac{\partial}{\partial \xi} \left(\sqrt{g_o} H u S \right) + \frac{\partial}{\partial \eta} \left(\sqrt{g_o} H v S \right) \right] \\
& + \frac{E_h}{S_{c_H}\sqrt{g_o}} \left[\frac{\partial}{\partial \xi} \left(\sqrt{g_o} H g^{11} \frac{\partial S}{\partial \xi} + \sqrt{g_o} H g^{12} \frac{\partial S}{\partial \eta} \right) \right] \\
& + \frac{E_h}{S_{c_H}\sqrt{g_o}} \left[\frac{\partial}{\partial \eta} \left(\sqrt{g_o} H g^{21} \frac{\partial S}{\partial \xi} + \sqrt{g_o} H g^{22} \frac{\partial S}{\partial \eta} \right) \right]
\end{aligned} \quad (2.21)$$

The tensor form of the governing equations can be found in Appendix B. Details on the turbulence scheme, used by the CH3D model to calculate the turbulent eddy viscosities and diffusivities, can be found in Appendix C.

2.2 Boundary Conditions

The boundary conditions at the free surface in a non-dimensional, vertically stretched ($\sigma = 0$), boundary-fitted coordinate system are

$$
\begin{array}{rcl}
A_V \frac{\partial u}{\partial \sigma} & = & \frac{H}{E_p} \tau_{s\xi}^w \\
A_V \frac{\partial v}{\partial \sigma} & = & \frac{H}{E_v} \tau_{s\eta}^w \\
\frac{\partial S}{\partial \sigma} & = & 0
\end{array}
\tag{2.22}
$$

The Cartesian wind stress, τ^w, is calculated using

$$
\begin{array}{rcl}
\tau_x^w & = & \rho_a C_{ds} u_w W_s \\
\tau_y^w & = & \rho_a C_{ds} v_w W_s
\end{array}
\tag{2.23}
$$

where $W_s = \sqrt{u_w^2 + v_w^2}$ is the total wind speed. The drag coefficient, C_{ds} is calculated using the relation developed by Garratt (1977)

$$
C_{ds} = 0.001 \times (0.75 + 0.067 W_s)
\tag{2.24}
$$

Boundary conditions, if specified in a Cartesian coordinate system, such as wind stress, must first be transformed before being used in the boundary-fitted equations. For example, the surface stress in the transformed system is given by

$$
\begin{array}{rcl}
\tau_{s\xi} & = & \frac{\partial \xi}{\partial x} \tau_{sx} + \frac{\partial \xi}{\partial y} \tau_{sy} \\
\tau_{s\eta} & = & \frac{\partial \eta}{\partial x} \tau_{sx} + \frac{\partial \eta}{\partial y} \tau_{sy}
\end{array}
\tag{2.25}
$$

The second surface boundary condition is the kinematic free surface boundary condition which states

$$
w = \frac{\partial \zeta}{\partial t} + u \frac{\partial \zeta}{\partial x} + v \frac{\partial \zeta}{\partial y}
\tag{2.26}
$$

The boundary conditions at the bottom in non-dimensional, vertically stretched ($\sigma = -1$), boundary-fitted coordinate system are

$$
\begin{array}{rcl}
A_V \frac{\partial u}{\partial \sigma} & = & \frac{H}{E_p} \tau_{b\xi}^w \\
& = & \frac{U_r}{A_{vr}} H_r Z_r C_d [g_{11} u_b^2 + 2 g_{12} u_b v_b + g_{22} v_b^2] u_b \\
A_V \frac{\partial v}{\partial \sigma} & = & \frac{H}{E_v} \tau_{b\eta}^w \\
& = & \frac{U_r}{A_{vr}} H_r Z_r C_d [g_{11} u_b^2 + 2 g_{12} u_b v_b + g_{22} v_b^2] v_b \\
\frac{\partial S}{\partial \sigma} & = & 0
\end{array}
\tag{2.27}
$$

where u_b and v_b are the contravariant velocity components at the first grid point above the bottom. The drag coefficient, C_d, is defined using the formulation of

Sheng (1983) which states that the coefficient is a function of the size of the bottom roughness elements, z_o, and the height at which u_b is measured, so long as z_1 is within the constant flux layer above the bottom. The size of the bottom roughness elements can be related to the Nikuradse equivalent sand gain roughness, k_s, using the relation $z_o = k_s/30$. The drag coefficient is defined as

$$C_d = \left(\frac{\kappa}{ln(z_1/z_0)}\right)^2 \tag{2.28}$$

where $\kappa = 0.4$ is the von Karman constant. Examples of how the drag coefficient varies with bottom roughness and measurement height are shown in Table 2.1.

Table 2.1: Drag coefficient, C_d, as a function of of bottom roughness, z_o, and measurement height, z_1.

	$z_1 = 10$ cm	$z_1 = 25$ cm	$z_1 = 50$ cm	$z_1 = 100$ cm
z_o=0.2 cm	0.0105	0.0069	0.0052	0.0041
z_o=0.4 cm	0.0154	0.0094	0.0069	0.0052
z_o=0.8 cm	0.0251	0.0135	0.0094	0.0069
z_o=1.0 cm	0.0302	0.0154	0.0105	0.0075
z_o=1.5 cm	0.0445	0.0202	0.0130	0.0091
z_o=2.0 cm	0.0618	0.0251	0.0154	0.0105

Along the shoreline where river inflow may occur, the conditions are generally

$$\begin{aligned} u &= u(x,y,\sigma,t) \\ v &= v(x,y,\sigma,t) \\ \omega &= 0 \\ S &= S(x,y,\sigma,t) \end{aligned} \tag{2.29}$$

Along solid boundaries, the no-slip condition dictates that the tangential velocity is zero while the slip condition requires that the normal velocity is zero. When flow is specified at a boundary, the normal velocity component is prescribed. Tidal boundary conditions are specified using water level, ζ, directly. During an ebb tide, the concentrations of salinity flowing out are calculated using a 1D advection equation while during a flood tide the offshore concentration is generally prescribed as either fixed or time varying.

2.3   Initial Conditions

To initiate a simulation, the initial spatial distribution of ζ, u, v, w, and S need to be specified. When these variables are unknown, "zero" initial fields can be used. When these variables are known at a number of locations, an initial field can be generated by performing spatial interpolation. It is desirable that an interpolated field satisfies the conservation equation of a particular variable.

For practical simulations, a "spin-up" period is required to damp out the transients caused by the initial condition which contains uncertainties regardless if real data were used to produce the initial condition. Spin-up refers to a simulation conducted for a time period before the time period of interest. The length of the spin-up period is variable and depends on such factors as basin size and current velocities. For Indian River Lagoon, the spin-up period is on the order of 5 to 10 days for barotropic simulations and 30 to 90 days for baroclinic simulations. The baroclinic spin-up period is longer than the barotropic spin-up period because salinity responds much more slowly than water levels and currents.

2.4   Finite Difference Equations

A space-staggered grid system is used to discretize the differential equations (Sheng, 1983). In the staggered grid system, elevation, vertical velocity, salinity and density are defined at the center of a cell, the u-velocity at the left and right sides of a cell, and v-velocity at the top and bottom sides of a cell (Figure 2.3). A staggered grid system has an accuracy of $O(\Delta x^2)$ as opposed to an accuracy of $O(\Delta x)$ in a non-staggered grid system.

Because of the staggered grid system, three types of sweeps are needed to solve the resulting finite difference equations: i-sweeps (Figure 2.4), j-sweeps (Figure 2.5) and k-sweeps (Figure 2.6). An i-sweep refers to the solution of a finite difference equation at u-nodes along a line of varying i's and a constant j while a j-sweep refers
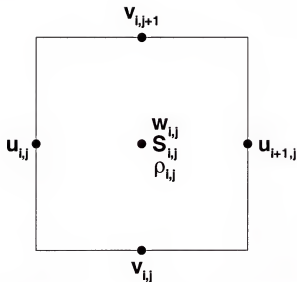
Figure 2.3: The staggered grid system used in the CH3D model.
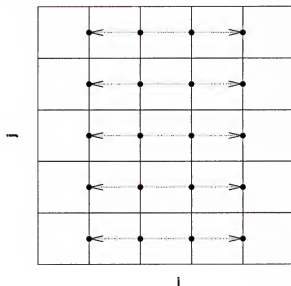
Figure 2.4: The i-sweeps in a 5×5 closed basin grid system.

to the solution of a finite difference equation at v-nodes along a line of varying j's and a constant i. A k-sweep refers to the solution of a finite difference equation at the center of cells along a line of varying i's and a constant j.
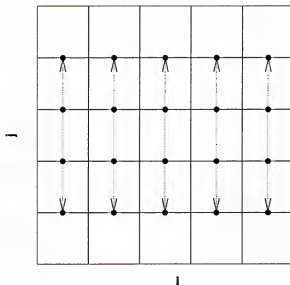
Figure 2.5: The j-sweeps in a 5×5 closed basin grid system.
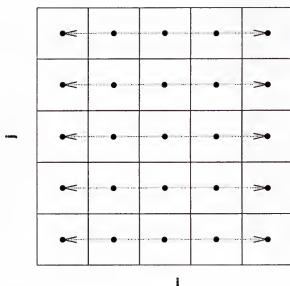
Figure 2.6: The k-sweeps in a 5×5 closed basin grid system.

The method used to solve the full three-dimensional equations is divided into two parts. The first part, the "external mode", solves the vertically integrated equations of motion and continuity over the entire computational domain. The second

part, the "internal mode", solve the three-dimensional equations of motion, continuity and transport in the water column for a given cell. Further details on the mode splitting method can be found in Sheng (1983) and Sheng (1987).

2.4.1 External Mode

In the boundary-fitted curvilinear coordinate system, the two-dimensional, non-dimensional, vertically integrated equations of motion, written in terms of contravariant velocity components, in the transformed coordinates, ξ and η, are:

$$\frac{\partial \zeta}{\partial t} + \frac{\beta}{\sqrt{g_o}}\left[\frac{\partial}{\partial \xi}(\sqrt{g_o}U) + \frac{\partial}{\partial \eta}(\sqrt{g_o}V)\right] = 0 \tag{2.30}$$

$$\begin{aligned}
\frac{\partial U}{\partial t} = \ &-H(g^{11}\frac{\partial \zeta}{\partial \xi} + g^{12}\frac{\partial \zeta}{\partial \eta}) - (\frac{g_{12}}{\sqrt{g_o}}U + \frac{g_{22}}{\sqrt{g_o}}V) \\
&+ \frac{R_o}{g_o}\int_{-1}^{0}[\xi-\text{Direction Inertia Terms}]d\sigma \\
&+ \tau_{s\xi} - \tau_{b\xi} - \frac{HR_o}{F^2}\int_{-1}^{0}[\xi-\text{Direction Baroclinic Terms}]d\sigma \\
&+ HE_H A_H \int_{-1}^{0}[\xi-\text{Direction Horizontal Diffusion Terms}]d\sigma
\end{aligned} \tag{2.31}$$

$$\begin{aligned}
\frac{\partial V}{\partial t} = \ &-H(g^{21}\frac{\partial \zeta}{\partial \xi} + g^{22}\frac{\partial \zeta}{\partial \eta}) - (\frac{g_{11}}{\sqrt{g_o}}U + \frac{g_{21}}{\sqrt{g_o}}V) \\
&- \frac{R_o}{g_o}\int_{-1}^{0}[\eta-\text{Direction Inertia Terms}]d\sigma \\
&+ \tau_{s\eta} - \tau_{b\eta} - \frac{HR_o}{F^2}\int_{-1}^{0}[\eta-\text{Direction Baroclinic Terms}]d\sigma \\
&+ HE_H A_H \int_{-1}^{0}[\eta-\text{Direction Horizontal Diffusion Terms}]d\sigma
\end{aligned} \tag{2.32}$$

The surface slope in the opposite direction ($Hg^{12}\frac{\partial \zeta}{\partial \xi}$ and $Hg^{21}\frac{\partial \zeta}{\partial \eta}$), nonlinear, diffusion, baroclinic, Coriolis, wind and bottom stress terms in Equations 2.31 and 2.32 are then lumped into the single terms F_ξ and F_η, yielding

$$\frac{\partial U}{\partial t} + Hg^{11}\frac{\partial \zeta}{\partial \xi} + F_\xi = 0 \tag{2.33}$$

$$\frac{\partial V}{\partial t} + Hg^{22}\frac{\partial \zeta}{\partial \eta} + F_\eta = 0 \tag{2.34}$$

where F_ξ and F_η are the remaining nonlinear, diffusion, baroclinic, Coriolis, wind and bottom stress terms. These simplified differential equations can then be written in the following finite difference form:

$$\begin{aligned}
\zeta_{i,j}^{n+1} &+ \frac{\theta \beta \Delta t}{\sqrt{(g_o)_{i,j,s}}}\left(\sqrt{(g_o)_{i+1,j,u}}U_{i+1,j}^{n+1} - \sqrt{(g_o)_{i,j,u}}U_{i,j}^{n+1}\right) \\
&+ \frac{\theta \beta \Delta t}{\sqrt{(g_o)_{i,j,s}}}\left(\sqrt{(g_o)_{i,j+1,v}}V_{i,j+1}^{n+1} - \sqrt{(g_o)_{i,j,v}}V_{i,j}^{n+1}\right) \\
&= \zeta_{i,j}^{n} - \frac{(1-\theta)\beta \Delta t}{\sqrt{(g_o)_{i,j,s}}}\left(\sqrt{(g_o)_{i+1,j,u}}U_{i+1,j}^{n} - \sqrt{(g_o)_{i,j,u}}U_{i,j}^{n}\right) \\
&- \frac{(1-\theta)\beta \Delta t}{\sqrt{(g_o)_{i,j,s}}}\left(\sqrt{(g_o)_{i,j+1,v}}V_{i,j+1}^{n} - \sqrt{(g_o)_{i,j,v}}V_{i,j}^{n}\right)
\end{aligned} \tag{2.35}$$

$$U_{i,j}^{n+1}\left(1+\theta_1\Delta t C_d\left|\overline{V}_{i,j,u}^n\right|\right)+\theta\Delta t g_{i,j,u}^{11}H(\zeta_{i,j}^{n+1}-\zeta_{i-1,j}^{n+1})$$
$$=U_{i,j}^n\left(1-\Delta t(1-\theta_1)C_d\left|\overline{V}_{i,j,u}^n\right|\right)-(1-\theta)\Delta t g_{i,j,u}^{11}H(\zeta_{i,j}^n-\zeta_{i-1,j}^n)-\Delta t F_{u,i,j} \quad (2.36)$$

$$V_{i,j}^{n+1}\left(1+\theta_1\Delta t C_d\left|\overline{V}_{i,j,v}^n\right|\right)+\theta\Delta t g_{i,j,v}^{22}H(\zeta_{i,j}^{n+1}-\zeta_{i,j-1}^{n+1})$$
$$=V_{i,j}^n\left(1-\Delta t(1-\theta_1)C_d\left|\overline{V}_{i,j,v}^n\right|\right)-(1-\theta)\Delta t g_{i,j,v}^{22}H(\zeta_{i,j}^n-\zeta_{i,j-1}^n)-\Delta t F_{v,i,j} \quad (2.37)$$

It is noted that these three governing equations for the external mode can be solved in two fundamentally different ways. The first method (Method I), developed for use in the original CH3D model (Sheng, 1987), solves the three governing hydrodynamic equations in two sweeps, an i-sweep and a j-sweep.

First, the external mode i-sweep combines the continuity equation and the u-momentum equation into two coupled equations, which when solved, yields an intermediate surface elevation, ζ^*, and the next time levels integrated velocity in the u-direction, U^{n+1}. Then, the j-sweep combines the continuity equation and the v-momentum equation into two coupled equations, which when solved using ζ^* and U^{n+1} calculated previously, yields the next time levels surface elevation, ζ^{n+1}, and the integrated velocity in the v-direction, V^{n+1}. When the finite difference form of these equations are solved, any given i-sweep is independent of any other i-sweep and any given j-sweep is independent of any other j-sweep; however, all i-sweeps must be solved before any j-sweep can be solved. Because of the independent nature of the sweeps, this method allows for relatively easy parallelization (Davis and Sheng, 2000).

In the second method (Method II), the u- and v-momentum equations (Equations 2.36 and 2.37) are substituted into the continuity equation (Equations 2.35). The resulting equation is a large multi-diagonal system of equations with the surface elevations as the unknowns. This method, when applied to the vertically averaged form of the equations, allows for relatively easy treatment of wetting and drying of the computational domain (Davis, 1996).

For an $n \times n$ grid, Method I results in $2n$ systems of $2n$ linear equations which again can be solved in $O(n^2)$ time. Method II results in a system of n^2 linear

equations which can be solved in $O(n^2)$ time. Thus, both methods can be solved in asymptotically the same time. Method I is used in the present study because Method II has not been fully incorporated into CH3D yet.

While neither method is asymptotically faster than the other, in terms of parallelization, Method I is slightly more advantageous in that the coupled equations themselves can be solved in parallel and/or the sweeps in a given direction can be solved in parallel whereas Method II only allows for parallelization of the solving of the multi-diagonal equations. The finite-difference equations for the i-sweep of Method I are given in the following.

i-sweep

The finite difference equations for the i-sweep consist of the two following equations for ζ^* and U^{n+1}.

$$\zeta_{i,j}^* + \frac{\theta\beta\Delta t}{\sqrt{(g_o)_{i,j,s}}} \left(\sqrt{(g_o)_{i+1,j,u}} U_{i+1,j}^{n+1} - \sqrt{(g_o)_{i,j,u}} U_{i,j}^{n+1} \right)$$
$$= \zeta_{i,j}^n - \frac{(1-\theta)\beta\Delta t}{\sqrt{(g_o)_{i,j,s}}} \left(\sqrt{(g_o)_{i+1,j,u}} U_{i+1,j}^n - \sqrt{(g_o)_{i,j,u}} U_{i,j}^n \right)$$
$$- \frac{\beta\Delta t}{\sqrt{(g_o)_{i,j,s}}} \left(\sqrt{(g_o)_{i,j+1,v}} V_{i,j+1}^n - \sqrt{(g_o)_{i,j,v}} V_{i,j}^n \right) \tag{2.38}$$

$$U_{i,j}^{n+1} \left(1 + \theta_1 \Delta t C_d \overline{|V_{i,j,u}^n|} \right) + \theta\Delta t g_{i,j,u}^{11} H(\zeta_{i,j}^* - \zeta_{i-1,j}^*)$$
$$= U_{i,j}^n - \Delta t (1 - \theta_1) C_d \overline{|V_{i,j,u}^n|} U_{i,j}^n - (1-\theta)\Delta t g_{i,j,u}^{11} H(\zeta_{i,j}^n - \zeta_{i-1,j}^n) - \Delta t F_{u,i,j} \tag{2.39}$$

The variables θ and θ_1 determine the degree to which certain terms are explicitly or implicitly treated. For example, if θ equals 1, then the surface elevation terms will be treated implicitly, if it is 0, the surface elevation terms will be treated explicitly, and if it is 0.5 they will be treated semi-implicitly. In a similar manner, θ_1 determines the degree to which the bottom friction terms are evaluated.

Rearranging Equation 2.38 gives a tridiagonal set of equations of the form:

$$a1_{i,j} U_{i,j}^{n+1} + b1_{i,j} \zeta_{i,j}^* + c1_{i,j} U_{i+1,j}^{n+1} = d1_{i,j} \tag{2.40}$$

where

$$a1 = -\frac{\theta\beta\Delta t \sqrt{(g_o)_{i,j,u}}}{\sqrt{(g_o)_{i,j,s}}} \tag{2.41}$$

$$b1 = 1 \tag{2.42}$$

$$c1 = \frac{\theta \beta \Delta t \sqrt{(g_o)_{i+1,j,u}}}{\sqrt{(g_o)_{i,j,s}}} \tag{2.43}$$

and

$$
\begin{aligned}
d1 = \ & \zeta_{i,j}^n - (1-\theta)\frac{\beta \Delta t}{\sqrt{(g_o)_{i,j,s}}}\left(\sqrt{(g_o)_{i+1,j,u}}U_{i+1,j}^n - \sqrt{(g_o)_{i,j,u}}U_{i,j}^n\right) \\
& -\frac{\beta \Delta t}{\sqrt{(g_o)_{i,j,s}}}\left(\sqrt{(g_o)_{i,j+1,v}}V_{i,j+1}^n - \sqrt{(g_o)_{i,j,v}}V_{i,j}^n\right)
\end{aligned}
\tag{2.44}
$$

The third subscript on the determinant of the metric tensor, g_o, indicates at which node in the computational grid the determinant is to be evaluated.

Rearranging Equation 2.39 also gives a tridiagonal set of equations. These equations take the form

$$a2_{i,j}\zeta_{i-1,j}^\star + b2_{i,j}U_{i,j}^{n+1} + c2_{i,j}\zeta_{i,j}^\star = d2_{i,j} \tag{2.45}$$

where

$$a2 = -\theta \Delta t g_{i,j,u}^{11} H_{i,j,u}^n \tag{2.46}$$

$$b2 = 1 + \theta_1 C_d |\overline{V}_{i,j,u}^n| \Delta t \tag{2.47}$$

$$c2 = \theta \Delta t g_{i,j,u}^{11} H_{i,j,u}^n \tag{2.48}$$

and

$$
\begin{aligned}
d2 = \ & U_{i,j}^n \left(1 - \Delta t(1-\theta_1)C_d|\overline{V}_{i,j,u}^n|\right) \\
& -(1-\theta)\Delta t g_{i,j,u}^{11} H_{i,j,u}^n(\zeta_{i,j}^n - \zeta_{i-1,j}^n) - \Delta t F_{u,i,j}^n
\end{aligned}
\tag{2.49}
$$

The third subscript on the depth, H, and the covariant metric tensor, g^{11}, also indicates at which node in the computational grid the values are to be evaluated.

Equations 2.40 and 2.45 are solved simultaneously for the intermediate water surface elevation, $\zeta^\star$, and the vertically integrated U^{n+1} velocity.

j-sweep

The finite difference equations for the j-sweep consist of the following two equations for ζ^{n+1} and V^{n+1}:

$$
\begin{aligned}
\zeta_{i,j}^{n+1} \ & + \frac{\theta \beta \Delta t}{\sqrt{(g_o)_{i,j,s}}}\left(\sqrt{(g_o)_{i,j+1,v}}V_{i,j+1}^{n+1} - \sqrt{(g_o)_{i,j,v}}V_{i,j}^{n+1}\right) \\
& = \zeta_{i,j}^\star - \frac{\theta \beta \Delta t}{\sqrt{(g_o)_{i,j,s}}}\left(\sqrt{(g_o)_{i,j+1,v}}V_{i,j+1}^n - \sqrt{(g_o)_{i,j,v}}V_{i,j}^n\right)
\end{aligned}
\tag{2.50}
$$

$$V_{i,j}^{n+1}\left(1 + \theta_1 \Delta t C_d \overline{|V_{i,j,v}^{*,n}|}\right) + \theta \Delta t g_{i,j,v}^{22} H(\zeta_{i,j}^{n+1} - \zeta_{i,j-1}^{n+1}) =$$
$$V_{i,j}^n - \Delta t(1 - \theta_1) C_d \overline{|V_{i,j,v}^{*,n}|} V_{i,j}^n - (1 - \theta)\Delta t g_{i,j,v}^{22} H(\zeta_{i,j}^* - \zeta_{i,j-1}^*) - \Delta t F_{n,i,j} \tag{2.51}$$

Rearranging Equation 2.50 gives a tridiagonal set of equations of the form:

$$a3_{i,j} V_{i,j}^{n+1} + b3_{i,j} \zeta_{i,j}^{n+1} + c3_{i,j} V_{i,j+1}^{n+1} = d3_{i,j} \tag{2.52}$$

where

$$a3 = -\frac{\theta \beta \Delta t \sqrt{(g_o)_{i,j,v}}}{\sqrt{(g_o)_{i,j,s}}} \tag{2.53}$$

$$b3 = 1 \tag{2.54}$$

$$c3 = \frac{\theta \beta \Delta t \sqrt{(g_o)_{i,j+1,v}}}{\sqrt{(g_o)_{i,j,s}}} \tag{2.55}$$

and

$$d3 = \zeta_{i,j}^* - \frac{\theta \beta \Delta t}{\sqrt{(g_o)_{i,j,s}}}\left(\sqrt{(g_o)_{i,j+1,v}} V_{i,j+1}^n - \sqrt{(g_o)_{i,j,v}} V_{i,j}^n\right) \tag{2.56}$$

Again, the third subscript on the determinant of the metric tensor, g_o, indicates the location of the node at which the determinant is to be evaluated.

Rearranging Equation 2.51 also gives a tridiagonal set of equations. These equations take the form

$$a4_{i,j} \zeta_{i-1,j}^{n+1} + b4_{i,j} V_{i,j}^{n+1} + c4_{i,j} \zeta_{i,j}^{n+1} = d4_{i,j} \tag{2.57}$$

where

$$a4 = -\theta \Delta t g_{i,j,v}^{22} H_{i,j,v}^* \tag{2.58}$$

$$b4 = 1 + \theta_1 C_d \overline{|V_{i,j,v}^{*,n}|} \Delta t \tag{2.59}$$

$$c4 = \theta \Delta t g_{i,j,v}^{22} H_{i,j,v}^* \tag{2.60}$$

and

$$d4 = V_{i,j}^n\left(1 - \Delta t(1 - \theta_1) C_d \overline{|V_{i,j,v}^{*,n}|}\right)$$
$$- (1 - \theta)\Delta t g_{i,j,v}^{22} H_{i,j,v}^*(\zeta_{i,j}^* - \zeta_{i,j-1}^*) - \Delta t F_{\eta,i,j}^n \tag{2.61}$$

The third subscript on the depth, H, and the covariant metric tensor, g^{22}, also indicates at which node in the computational grid the values are to be evaluated.

Equations 2.52 and 2.57 are solved simultaneously for the new water surface elevation, ζ^{n+1}, and the vertically integrated velocity, V^{n+1}.

2.4.2   Internal Mode and Salinity Transport

In the internal mode, the layer velocities are calculated using the updated integrated velocities and the layer velocities at the previous time step. The vertical diffusion is treated implicitly while all other terms are calculated explicitly. The u-layer velocities are solved using the i-sweeps while the v-layer velocities are solved using the j-sweeps, both described previously. A more detailed explanation of the internal mode including both differential as well as finite difference equations appears in Appendix D.

Salinity transport is calculated in a similar manner to the internal mode. Vertical diffusion is calculated implicitly while the remaining advection and horizontal diffusion are treated explicitly. Because the salinity is defined at the center of a computation cell, the salinity at the new time step is calculated using a series of k-sweeps. Details on the finite difference equation and solution algorithms for salinity transport can be found in Appendix E.

2.5   Model Domain and Grid Systems

The model domain is approximately 230 km long and extends from Ponce de Leon Inlet in the north to St. Lucie Inlet in the south. The model domain is extended south of the SJRWMD boundary (Figure 2.7) to allow easier specification of the model boundary conditions.

2.5.1   Boundary-fitted Grids

The northern boundary of the model grid is positioned slightly to the north of Ponce de Leon Inlet and the southern boundary is positioned slightly to the south of St. Lucie Inlet. The western boundary includes as many of the major rivers and canals as practical and the eastern boundary is placed at the four tidal inlets present in the model domain. From north to south, these inlets are Ponce de Leon, Sebastian,

Ft. Pierce and St. Lucie. The interior coastline is then fitted with a boundary-fitted grid which is non-orthogonal but as orthogonal as possible.

As was stated previously, for this study, the eastern boundary is placed at the four tidal inlets. The advantage of placing the boundary directly at the inlets is because the availability of measured water level data at the inlets (as will be discussed later) allows for relatively easy specification of water level boundary conditions. This is very important for the simulation of water level and currents within the lagoon. The main disadvantage arises when simulating conservative species (such as salinity) within the lagoon, specifically, how the tidal boundary condition is applied to the species. During an ebb tide, the concentrations of species flowing out can be calculated using a 1D advection equation; however, during a flood tide the offshore concentration is generally unknown; thus, preventing the use of a 1D advection equation. Hence, some value will need to be chosen as the inflow concentration. If the eastern boundary was placed well offshore, specification of this inflow concentration becomes easier because conservative species concentrations become more uniform and their affect upon the interior of the estuary become less.

Two sets of non-orthogonal, boundary-fitted grids in the horizontal plane were developed for the IRL-PLR model. The first boundary-fitted grid, developed by Sheng et al. (1996a), was generated using the WESCORA software package developed by Thompson (1985) and then further refined through various techniques for this study (Figure 2.7). This grid is (477 × 43) and has an average cell width of 434 m and will be referred to as the "fine grid" from here on. A "coarse grid", which is generated by modifying and removing cells in the "fine grid", is (198 × 22) and has an average cell width of 940 m (Figure 2.8). Characteristics of both the coarse and fine grids are shown in Table 2.2. Unless otherwise noted, all simulations presented herein were performed using the fine grid in the horizontal plane and in the vertical direction, four vertical layers with a uniformly spaced σ-grid.
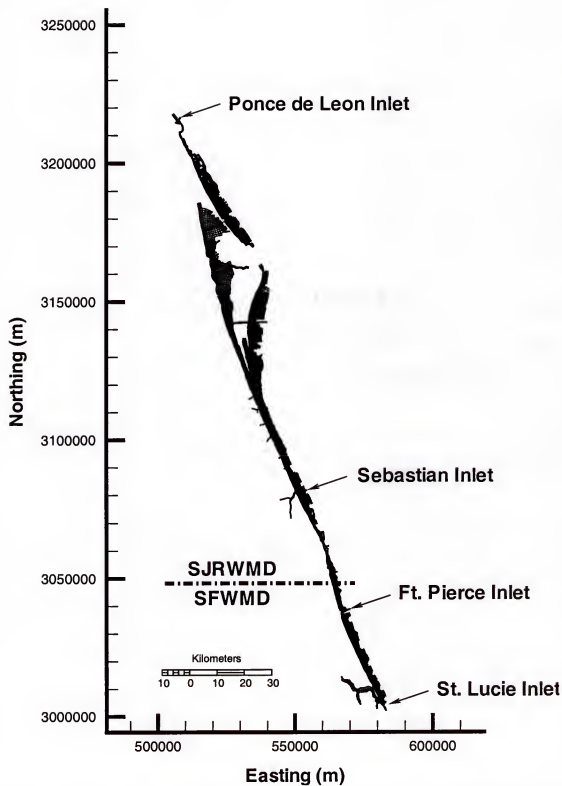
Figure 2.7: Boundary-fitted "fine grid" (477×43) used for numerical simulations of the IRL.
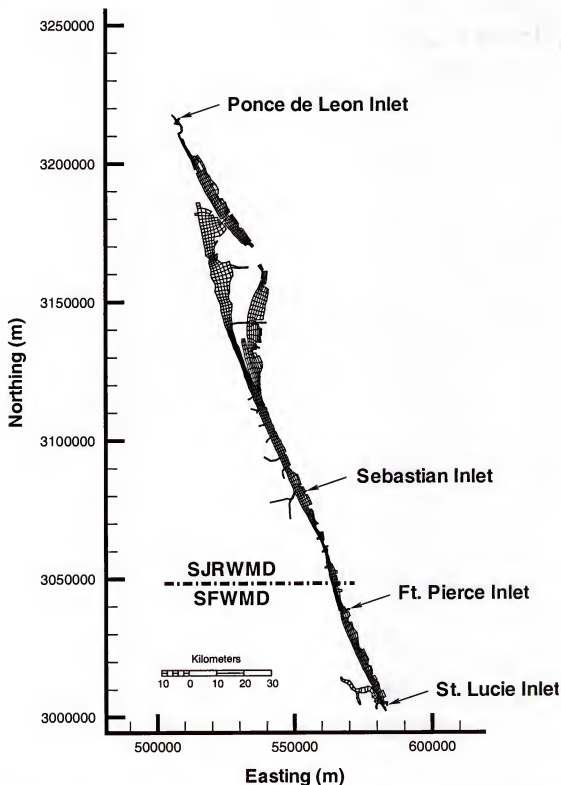
Figure 2.8: Boundary-fitted "coarse grid" (198 × 22) used for numerical simulations of the IRL.

Table 2.2: Characteristics of the IRL coarse and fine grids in the horizontal plane.

Parameter	coarse grid	fine grid
Grid size	198×22	477×43
Total number of cells	4356	20511
Number of water cells	1184	4921
Percentage of water cells	27	24
Minimum cell width (m)	52	45
Maximum cell width (m)	3654	1527
Average cell width (m)	940	434
Average cell aspect ratio (i-direction/j-direction)	2.0	1.4
Minimum cell angle (deg)	13	35
Maximum cell angle (deg)	191	153
Average minimum cell angle (deg)	75	83
Average maximum cell angle (deg)	105	98
Percentage of cells which contain an angle less than 45 degrees	2.7	0.1
Minimum cell depth (cm NAVD88)	20	13
Maximum cell depth (cm NAVD88)	1091	998
Average cell depth (cm NAVD88)	197	203

During the study, several combined inshore and offshore grid systems were created. The combined grid systems alleviated problems associated with applying hydrodynamic and salinity transport directly at the inlets by moving the boundary conditions offshore. However, new difficulties were created when trying to specify water level and salinity along a long offshore boundary. Additionally, these grid systems had as many as $2\times$ the number of computational cells as the fine grid and therefore the simulations took twice as long to perform. Because a one year simulation of hydrodynamics and salinity transport took several days, the additional computational time required made simulations not feasible for this study.

Numerous geographic features which restrict flow are present throughout the IRL. While most of the large scale features have been resolved with the boundary-fitted grid, smaller features, such as causeways, were resolved using thin-wall barriers. These barriers block flow through one side of a grid cell allowing long, thin features

to be easily resolved. Appendix G shows the locations of the 16 thin-wall barrier resolved causeways within the IRL fine grid.

2.5.2 Bathymetry

Using bathymetry provided by the SJRWMD, the coarse and fine grid bathymetries (Figure 2.9) were developed. While an inverse distance interpolation followed by a simple smoothing scheme was the primary method of determining bathymetry in the lagoon, to ensure the proper passage of flow through various restrictions in the lagoon, several key areas had their bathymetry further adjusted after the interpolation and smoothing was performed. Rather than modifying the depth directly, cross-sectional areas in the boundary-fitted grid system were matched to those of measured data. A list of the key locations modified is shown in Table 2.3.
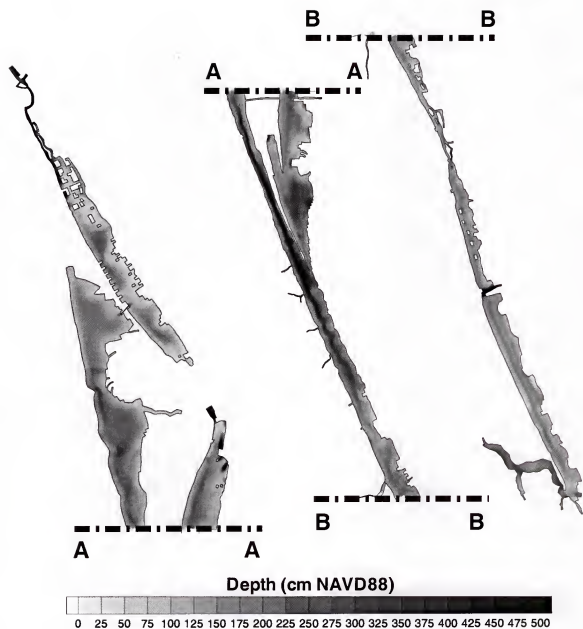
Figure 2.9: IRL bathymetry in the fine grid.

Table 2.3: Cross-sectional areas at key flow restriction points within the lagoon. The causeways cross-sectional areas marked with a "*" were not actually modified by hand but are included so that the cross sectional areas at all of the resolved causeways can be shown.

Location	Longitude (deg W)	Latitude (deg N)	UTM X(m)	UTM Y(m)	Area (m^2 NAVD88)
Causeways					
Train	80 48 35	28 39 02	518621	3169304	1443
Brewer	80 48 02	28 37 10	519531	3165863	2211
NASA West	80 46 06	28 31 37	522676	3155659	2635
NASA East	80 36 44	28 30 48	537950	3154166	396
Bennett East	80 38 52	28 24 17	534507	3142139	788
Bennett West	80 44 18	28 24 08	525653	3141825	1968
Merritt	80 38 58	28 21 26	534373	3136859	563
Hubert Humphrey	80 43 08	28 21 20	527545	3136676	649
Pineda East	80 37 46	28 12 35	536368	3120534	604
Pineda West	80 39 13	28 12 20	534005	3120061	2526
Eau Gallie (East)	80 36 12	28 08 12	538950	3112465	686
Eau Gallie (West)	80 37 12	28 07 54	537334	3111887	2811
Melbourne	80 35 39	28 04 58	539875	3106472	2181
Wabasso	80 25 36	27 45 15	556520	3070160	926
North Beach	80 19 43	27 28 13	566356	3038764	1610
732 (West)	80 13 14	27 15 09	577179	3014682	1019*
732 (East)	80 12 48	27 15 16	577893	3014907	503*
A1A (at St. Lucie)	80 11 39	27 12 20	579829	3009509	823*
Inlets					
Ponce de Leon	80 54 53	29 03 46	508307	3214992	2322
Sebastian	80 26 55	27 51 36	554300	3081876	535
Ft. Pierce	80 17 57	27 28 05	569267	3038509	2200
St. Lucie	80 09 55	27 09 51	582721	3004930	1947
Other					
Haulover Canal	80 45 17	28 44 10	523957	3178825	201
Dragon Point	80 36 12	28 08 36	538953	3113180	765

CHAPTER 3
THE PARALLEL CH3D MODEL AND IRL GRID SYSTEM

In this chapter, the modification of the CH3D model to archive parallelism via the use of domain decomposition and shared memory techniques is presented. Additionally, four domain decomposition methods are explored and applied to the IRL grid system developed in Chapter 2.

In addition to the parallelization of the CH3D model presented herein, researchers have parallelized some of the other popular hydrodynamic models; however, this study is the only one to parallelize a fully coupled hydrodynamic, salinity, sediment and water quality model. Beck et al. (1995) parallelized the explicit Miami Isopycnic Coordinate Ocean Model[1] (MICOM) (Bleck et al., 1992; Bleck and Chassignet, 1994) using both shared memory and message passing techniques. Zhu et al. (1998a,b) parallelized the CH3D-WES hydrodynamic model by performing a simple 1-d decomposition of the grid system in the j-direction. The i-sweeps in the parallel CH3D-WES model are solved without modification while the j-sweeps are solved using message passing techniques. Luong et al. (2000) attempted to improve the poorly load-balanced parallel CH3D-WES model by dynamically threading the computation domains based on the amount of work available. Boukas et al. (2000) parallelized the explicit Princeton Ocean Model[2] (POM) (Blumberg and Mellor, 1980) using a message passing technique.

---

[1]http://panoramix.rsmas.miami.edu/micom/MICOM_contents.html

[2]http://www.aos.princeton.edu/WWWPUBLIC/htdocs.pom/

3.1   Parallel Approaches for the CH3D Model

The serial CH3D model can be converted into a parallel model using two possible approaches. The systems of equations in the sweeps can be calculated in parallel or the sweeps themselves can be split up and calculated in parallel.

3.1.1   Parallelizing the Individual Tridiagonal Systems

As described in the previous chapter, the model performs both external and internal mode calculations in sweeps. In the external mode, where the water level and integrated velocities are calculated, all i-sweeps, then all j-sweeps are performed. Equations 2.40 and 2.45 and Equations 2.52 and 2.57 are the coupled simultaneous tridiagonal systems of equations of the external mode. In the internal mode, where layer velocities and transport variables are calculated, vertically implicit equations are solved with no regard to the order in which the vertical columns are solved. CH3D solves the u- and v-layer velocities in i- and j-sweeps, respectively and the w-velocity, salinity and density in k-sweeps.

The first parallel approach is to use a parallel algorithm to solve the coupled tridiagonal systems of the external mode and then the implicit equations of the internal mode. Extensive testing determined that these matrices need to be of $O(1000)$ before parallelization becomes effective (see Appendix F on page 228). However, the average i- and j-sweep lengths of the IRL fine grid is $O(10)$ (Table 3.1) and for the internal mode, the number of vertical layers is typically $O(1)$. Because of the small average sweep length and number of vertical layers, parallelizing the individual equations in the external and internal modes would only be effective when applied to domains with a extremely large number of grid cells. The current serial model uses $O(100)$ MB of memory so a grid $100\times$ larger in each horizontal direction and $1000\times$ larger in the vertical direction would require $O(100,000)$ gigabytes of memory. Thus, the first parallel approach is deemed impractical for current applications.

Table 3.1: Sweep lengths in the boundary-fitted IRL fine grid.

Sweep Direction	Total Number of Sweeps	Average Sweep Length
i	351	13
j	673	6
k	484	10

3.1.2 Parallelizing the i-, j-, and k-sweeps

Due to the sweeping decomposition of the CH3D model, the i-, j-, and k-sweeps are independent of all other i-, j- and k-sweeps, respectively. This independence allows the i-, j- and k-sweeps to be divided among a group of processors and performed in parallel. Examples of an IRL fine grid i-, j- and k-sweep in the physical and computation domains are shown in Figures 3.1 and 3.2, respectively. This method proves to be effective (a reduction in model running time) in most cases where a minimal amount of work is done to overcome overhead associated with the parallelism.

When decomposing the individual sweeps, it very important that each processor perform the same amount of work. This concept is referred to as "load balancing". Four run-time methods were tested when splitting up the individual i-, j- and k-sweeps and are described as follows:

- I) Equal number of rows per processor (in order)

  The main advantage of this approach is its simplicity in application. The current model was optimized for a vector computer and hence the row sweeping system is in place, albeit not in parallel. For an n-processor machine, the first processor gets the first $1/n$ rows, the second processor, the second $1/n$ rows, and so on.
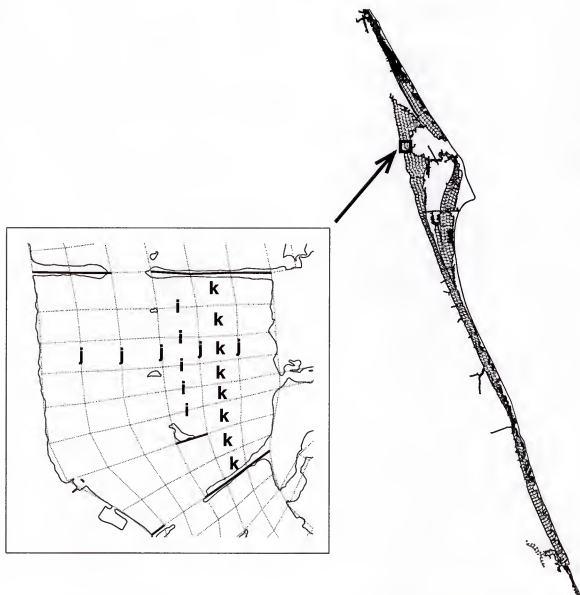
Figure 3.1: Examples of individual i, j and k sweeps within the IRL grid system. The shoreline is shown with thin solid lines, thin-wall barriers are shown with thick solid lines and the fine grid is shown with dotted lines.
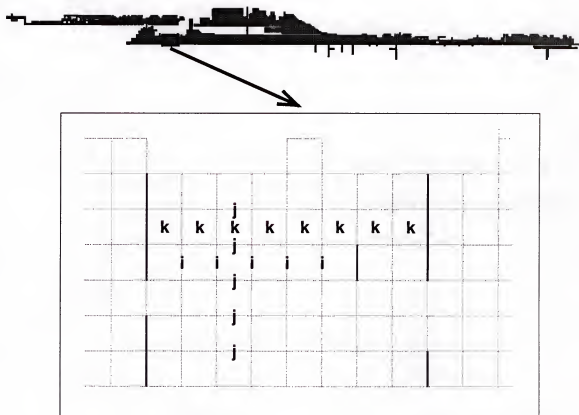
Figure 3.2: Examples of individual i, j and k sweeps within the transformed computational grid system. Thin-wall barriers are shown with thick solid lines and the transformed fine grid is shown with dotted lines.

In addition to being a relatively simple way to generate the sweeps, this method also has a memory advantage. Because sweeps are in order, the memory they address will also be in order and hence is more likely to be loaded on the processor. While memory transfer is very fast, repeated transfer of memory on and off the processor can slow down overall computational speed.

The main disadvantage of this method is the possibility of a disproportionate amount of CPU time being spent by one processor, in other words, poor load balancing. If one processor has 10 sweeps of average length 10 and another processor has 10 sweeps of average length 100, the latter processor will spend more time solving the tridiagonal matrices with the former processor will remain idle. For grid systems with a low percentage of water cells, like the IRL grid systems (Table 2.2), this method leads to very poor load balancing. Using this method, an example of the four processor mapping of the j- and j-sweeps to the IRL fine grid appears in Figure 3.3.

- II) Equal weight of rows per processor (in order)

  This method is similar to Method I except that a water cell weighting factor is introduced. Instead of an equal number of rows on each processor, each processor receives an equal number of water cells, such that the given cells constitute a complete row.

  While slightly more difficult in application, this method provides superior load balancing and retains the same memory advantage as discussed with Method I. Using this method, an example of the four processor mapping of the i- and j-sweeps to the IRL fine grid appears in Figure 3.4.

- III) Random rows per processor (in order)

  To offset the possible load balancing problems of Method I, a random ordering of sweeps can be used. While still having the same number of sweeps on each

Figure 3.3: The mapping of the IRL fine grid system onto 4 processors using Method I. The processor distribution of the a) *i*-sweep rows and the b) *j*-sweep columns are shown. All sweeps of a given color are solved by the same processor.

processor, the sweeps would no longer have to be in order. Using a probability analysis, it can be shown that with high probability each of the processor sweeps will have the same average length and hence total computation time on each processor will be the same. For a given processor, the sweeps it is given to solve are performed in order.

The downside of this method is that the sweeps are no longer in order. This causes the processors to load a new set of memory for each sweep. While memory transfer is fast, constantly transferring data in and out of the processor on such a large scale would slow the overall computation down. However,

Figure 3.4: The mapping of the IRL fine grid system onto 4 processors using Method II. The processor distribution of the a) i-sweep rows and the b) j-sweep columns are shown. All sweeps of a given color are solved by the same processor.

this method's improved load balancing would probably not offset any memory problems which may result.

Additionally, while a probability analysis will show that the work will be balanced, this is only true when averaged over a large number of simulations. For any given one simulation, the work may not be balanced. Using this method, an example of the four processor mapping of the i- and j-sweeps to the IRL fine grid appears in Figure 3.5.

- IV) Random rows per processor (random order)

Figure 3.5: The mapping of the IRL fine grid system onto 4 processors using Method III. The processor distribution of the a) i-sweep rows and the b) j-sweep columns are shown. All sweeps of a given color are solved by the same processor.

Like Method III each processor is assigned a random set of sweeps. These sweeps are then solved in a random fixed order determined at the onset of computation. Again, this method is designed to use randomness to provide better load balancing. However, now that the random sweeps are solved in random order, memory transfer issues increase. Using this method, an example of the four processor mapping of the i- and j-sweeps to the IRL "fine grid" (477x43) appears in Figure 3.6.

A series of one day simulations of hydrodynamics and salinity using the IRL fine grid (1400 time iterations using a time step of 60 s) were conducted and timed

Figure 3.6: The mapping of the IRL fine grid system onto 4 processors using Method IV. The processor distribution of the a) i-sweep rows and the b) j-sweep columns are shown. All sweeps of a given color are solved by the same processor.

to determining which method performed best (caused the largest reduction of model run time).

Method I is the easiest to implement; however, since the amount of work on each processor is not guaranteed to be the same, it did not perform the best. Method II performed the best of all the methods because the amount of work each processor does is split as evenly as possible, otherwise, the model parallel model will only be as fast as the processor which performs the most work. Method III and IV were designed to test the feasibility of randomized algorithms. These methods did not

perform as well as the first two, most likely because of the non-sequential memory access required and possible poor load balancing of the random algorithm.

In addition to the timing simulations, an analysis of the amount of work given to each processor was performed. Tables 3.2 and 3.3 show the amount of work each processor was assigned using the four methods for a typical four processor simulation. The standard deviation of the amount of work performed by each processor is used as an indicator of load balancing. The smaller the value, the more evenly the work is divided among the processors. This indicator confirms the conclusion of the parallel timing simulations, that Method II performs best in terms of load balancing. Hence, Method II will be used for all future parallel CH3D simulations.

Table 3.2: The amount of work (number of computation cells) performed by each processor during by an i-sweep of a simulation on the IRL fine grid.

Method	P_0	P_1	P_2	P_3	Standard Deviation
I	1466	1071	1128	767	286
II	1107	1107	1118	1100	7
III	1258	1028	921	1225	161
IV	1342	918	1258	914	224

Table 3.3: The amount of work (number of computation cells) performed by each processor during by a j-sweep of a simulation on the IRL fine grid.

Method	P_0	P_1	P_2	P_3	Standard Deviation
I	827	1351	1247	674	326
II	1021	1027	1027	1024	3
III	999	1056	1080	964	53
IV	984	1094	1114	907	97

3.2 Parallel Implementation

Shared memory parallelization is generally implemented with parallel constructs which are added to the serial source code. These constructs vary slightly

between compiler vendors but a standard called OpenMP[3] is now being supported by major compiler vendors (*e.g.* SGI, HP, Intel, Sun, Compaq, Absoft, and the Portland Group).

The application of the parallel constructs to the CH3D model, which is written in Fortran, revolves heavily around the use of macros and shell scripts (Davis and Sheng, 2000). These tools allow parallel constructs to be hidden within the code so as to make it more user friendly. The macros are also used to increase code portability because parallel constructs are implemented in slightly different methods under different compilers. Two different macro preprocessors are used in the development of the parallel CH3D code, they are

- *cpp* is the standard C preprocessor. It performs simple text substitutions, manipulations and conditional inclusion as described by the C standard.

- *m4* is a sophisticated macro processor intended as a front end for various programming languages.

Since the *m4* macros can span multiple lines while the *cpp* macros cannot, the *cpp* macros could be replaced by *m4* macros. However, to make the code clearer, *cpp* macros are used like flags which turn on or off sections of code while *m4* macros are only used to expand large multi-line text. The *cpp* macros work like "if" statements in Fortran; however, they are applied before compilation so that any code contained within a false if-block will not be compiled. In the parallel CH3D code, a programming convention has been established where the *cpp* macros are in upper case and begin and end with an underscore (_CPP_MACRO_EXAMPLE_) while the *m4* macros are lower case and begin and end with two underscores (__m4_macro_example__).

Listing 3.1 is a section of the combined serial and parallel source code for the CH3D salinity transport routine, "ch3dsa.f". The shaded text is additional lines of

---

[3]http://www.openmp.org

code added to the serial version (unshaded text) to produce the combined serial and parallel version. In total, 21 lines and 1 *cpp* variable, "_PARALLEL_CH3DSA_2_" are added to make this section of the salinity transport routine parallel. The word "PARALLEL" refers to the *cpp* variable being used as a parallel processing flag and "CH3DSA" refer to the name of the procedure being parallelized, "ch3dsa.f". Because there can be any number of parallel sections within a given routine, the "2" refers to the second section of parallel code section within "ch3dsa.f".

Listing 3.1: A section of combined serial and parallel source code for the salinity transport routine (ch3dsa.f). Shaded regions indicate the additional code needed for parallelization. The line numbers on the left are with respect to the transport routine.

```
940    #ifdef _PARALLEL_CH3DSA_2_
941
942    _parallel_start_
943
944    _parallel_var_ (shared, NUM_PROCESSORS, AHSN, SAI)
945    _parallel_var_ (shared, FXYZ, SA, GB, KS1, KS2, KROW, NS, AHSS, DTI)
946    _parallel_var_ (shared, PROC_GROUP_K_LENGTH, PROC_GROUP_K, DUM1, DZZ)
947    _parallel_var_ (shared, TIMES, IFC, SAO, DUM2, DUM3, COR)
948    _parallel_var_ (shared, SAL_ASSIMILATION_TERM)
949
950    _parallel_var_ (private, PAR_LOOP, L, L_TEMP, J, KK)
951    _parallel_var_ (private, TMTX1, TMTX2, TMTX3, TMTX4, CC,A1,A2,A3,A4)
952    _parallel_var_ (private, C1, C2, C3, C4, I, K, I1, I2, AGX)
953
954        do PAR_LOOP=1, NUM_PROCESSORS
955
956        do L = 1, PROC_GROUP_K_LENGTH(PAR_LOOP)
957
958            L_TEMP=PROC_GROUP_K(PAR_LOOP,L)
959
960            J  = KROW(L_TEMP)
961            I1 = KS1 (L_TEMP)
962            I2 = KS2 (L_TEMP)
963    #else
964        do L =  1, KKOL
965    C
966    C     Define the starting and ending values in each row
967    C
968            J  = KROW(L)
969            I1 = KS1(L)
970            I2 = KS2(L)
971    #endif
972    C
973    C Calculate advection and diffusion of salinity
974    C
```

```
975           do I=I1, I2
                  (code removed)
1040          end do
1041
1042       end do                          ! do L=1, ...
1043  #ifdef _PARALLEL_CH3DSA_2_
1044       end do                          ! do PAR_LOOP=1, NUM_PROCESSORS
1045  #endif
```

The k-sweeps are parallelized as follows (i- and j-sweeps are parallelized in a similar manner). Prior to time iteration, a routine is called ("ch3dsweep.f") which calculates the total weight of all the sweeps in the k-direction. This total weight is then divided by the number of processors, this value is the target weight. The routine then steps through the sweeps, summing the weights as it goes until the target weight is hit or exceeded. This weight is then compared to the weight with the last sweep removed, which ever row's weight, which when summed with the proceeding weights is closest to the target weight, is then set as the last sweep for the first processor. This process then repeats for the 2nd, 3rd, ... and n^{th} processors. The number of sweeps in a given direction for each processor is given by the variable "PROC_GROUP_K_LENGTH". The sweeps are numbered sequentially starting from 1. The variable "PROC_GROUP_K" is then calculated to give the corresponding sweep numbers that each processor is to work on.

In the actual loops to be parallelized, the loop that is made parallel is one which increments from 1 to the number of processors. This is the variable "PAR_LOOP". The variable "L" is then looped from 1 to the value of these variables for a given processor number. Using "L" and the processor number "PAR_LOOP", the variable "PROC_GROUP_K" then gives the sweep number that the given processor is to calculate.

Listing 3.2 is the expansion of the $m4$ macro "_parallel_start_" given on line 942 of Listing 3.1. Because the OpenMP shared memory programming standard is not fully supported by all compiler vendors yet, both Sun Microsystems and Cray

style parallel loop constructs are implemented using the *m4* macros. The *cpp* variables "_PARALLEL_OPENMP_", "_PARALLEL_SUNSTYLE_" and "_PARALLEL_CRAYSTYLE_" will activate the appropriate constructs when defined during compilation. The OpenMP construct "default(none)" (line number 3) removes all implicit declarations of variables within the loop. Normally, arrays are defined as "shared" variables and all other variables are "private" variables. Variables which are "shared" are seen identically by all processors within a parallel loop. Processors have their own copies of variables and are unaware of their counterparts on other processors when they are marked as "private". After the parallel loop has completed, the status of "private" variables are unknown. The "shared" and "private" declarations are independent of the type of variable (real, integer or logical).

Listing 3.2: Expansion of the *m4* macro "_parallel_start_".

```
1   #ifdef _PARALLEL_OPENMP_
2   !$omp parallel  do
3   !$omp& default(none)
4   #endif
5
6   #ifdef _PARALLEL_SUNSTYLE_
7   C$PAR DOALL MAXCPUS( _PROC_MAX_ )
8   #endif
9
10  #ifdef _PARALLEL_CRAYSTYLE_
11  !MIC$ DOALL MAXCPUS( _PROC_MAX_ )
12  #endif
```

Listing 3.3 is the expansion of the *m4* macro "_parallel_var_(private, PAR_LOOP, L, L_TEMP, J, KK)" given on line 944 of Listing 3.1. These lines declare the four variables "PAR_LOOP", "L", "L_TEMP", "J" and "KK" as "private" variables using the three different styles of parallel constructs.

Listing 3.3: Expansion of the *m4* macro "_parallel_var_(shared, NUM_PROCESSORS, AHSN, SAI)".

```
1   #ifdef _PARALLEL_OPENMP_
2   !$omp& shared(NUM_PROCESSORS,AHSN,SAI)
3   #endif
4
5   #ifdef _PARALLEL_SUNSTYLE_
6   C$PAR& ,shared(NUM_PROCESSORS,AHSN,SAI)
```

```
 7  #endif
 8
 9  #ifdef _PARALLEL_CRAYSTYLE_
10  !MIC$& ,shared(NUM_PROCESSORS,AHSN,SAI)
11  #endif
```

Listing 3.4 is the expansion of the *m4* macro

"_parallel_var_(shared, NUM_PROCESSORS, AHSN, SAI)" given on line 950 of

Listing 3.1. These lines declare the three variables "NUM_PROCESSORS", "AHSN"

and "SAI" as "shared" variables using the three different styles of parallel constructs.

Listing 3.4: Expansion of the *m4* macro "_parallel_var_(private, PAR_LOOP, L, L_TEMP, J, KK)".

```
 1  #ifdef _PARALLEL_OPENMP_
 2  !$omp& private(PAR_LOOP,L,L_TEMP,J,KK)
 3  #endif
 4
 5  #ifdef _PARALLEL_SUNSTYLE_
 6  C$PAR& ,private(PAR_LOOP,L,L_TEMP,J,KK)
 7  #endif
 8
 9  #ifdef _PARALLEL_CRAYSTYLE_
10  !MIC$& ,private(PAR_LOOP,L,L_TEMP,J,KK)
11  #endif
```

Listing 3.5 is the complete expansion of all the *m4* macros in Listing 3.1 along

with the *cpp* variables "_PARALLEL_CH3DSA_2_" and "_PARALLEL_OPENMP_"

being set. Only after this final expansion is the source code sent through the compiler.

Listing 3.5: The section of parallel source code after processing by *cpp* and *m4* . The *cpp* variables "_PARALLEL_CH3DSA_2_" and "_PARALLEL_OPENMP_" have been set. The line numbers on the left are with respect to the transport routine.

```
2917  !$omp parallel do !$omp& default(none)
2918
2919  !$omp& shared(NUM_PROCESSORS,AHSN,SAI)
2920  !$omp& shared(FXYZ,SA,GB,KS1,KS2,KROW,NS,AHSS,DTI)
2921  !$omp& shared(PROC_GROUP_K_LENGTH,PROC_GROUP_K,DUM1,DZZ)
2922  !$omp& shared(TIMES,IFC,SAO,DUM2,DUM3,COR)
2923  !$omp& shared(SAL_ASSIMILATION_TERM)
2924
2925  !$omp& private(PAR_LOOP,L,L_TEMP,J,KK)
2926  !$omp& private(TMTX1,TMTX2,TMTX3,TMTX4,CC,A1,A2,A3,A4)
2927  !$omp& private(C1,C2,C3,C4,I,K,I1,I2,AGX)
2928
2929      do PAR_LOOP=1, NUM_PROCESSORS
2930
```

```
2931        do L = 1, PROC_GROUP_K_LENGTH(PAR_LOOP)
2932
2933            L_TEMP=PROC_GROUP_K(PAR_LOOP,L)
2934
2935            J  = KROW(L_TEMP)
2936            I1 = KS1 (L_TEMP)
2937            I2 = KS2 (L_TEMP)
2938  C
2939  C Calculate advection and diffusion of  salinity
2940  C
2941            do I=I1, I2
                   (code removed)
3007            end do
3008
3009        end do                          ! do L=1, ...
3010
3011      end do                           ! do PAR_LOOP=1, NUM_PROCESSORS
```

The complete code assembly and compilation process takes the following form:

- Original source code (∗.f) is sent through the *m4* processor (∗.F)

- Resultant code (∗.F) is sent through the *cpp* processor (∗.i)

- Resultant code (∗.i) is sent through the compiler (∗.o)

- Object files (∗.o) are linked to form executable

By using macros, the CH3D model can be made parallel with a minimal amount of additional coding. The parallelism within procedures can be turned on or off with relative ease allowing for easy debugging and multiple construct formats can be supported with minimal cluttering of the original source code.

In total, 34 sections of code, in 24 separate procedures, were converted to the combined serial and parallel source code. These procedures comprise all current CH3D models, including: hydrodynamics and salinity, flushing, wave, sediment, nutrient, light and seagrass. This study presents results of the parallel hydrodynamics and salinity models, Du (2000) presented results using the parallel hydrodynamics,

salinity and flushing models and Sun (2001) presented results of the parallel hydrodynamics, salinity, wave and sediment models. Studies using the parallel nutrient, light and seagrass models will be completed soon.

CHAPTER 4
PARALLEL CH3D SIMULATION OF THE IRL

In this chapter, a numerical experiment is presented to demonstrate an application of the parallel CH3D model to the prediction of circulation and salinity transport in the IRL.

4.1 Overview of the 1998 Simulation

To study mechanisms responsible for longer-term regulation of hydrodynamics and salinity it is necessary to perform long-term simulations. To this end, year-long simulations of hydrodynamics and salinity transport were performed to

- further verify the hydrodynamic and salinity transport model,

- study seasonal variability of hydrodynamics and salinity transport, and

- study long-term water and salt budgets within the lagoon.

Due to the availability of measured data for model boundary conditions and model validation, calendar year 1998, 00:00 January 1st through 00:00 December 27th EST (360 days), was chosen as the time period for the year-long simulations. As will be discussed, boundary forcing for the 1998 simulations include observed water level, wind, fresh water inflow, precipitation, and evaporation.

4.2 Available Measured Data and Boundary Conditions

Hydrodynamic monitoring of the IRL has been conducted by a variety of organizations including the St. Johns River Water Management District[1] (SJRWMD), the

---

[1] http://sjr.state.fl.us/

South Florida Water Management District[2] (SFWMD), the United States Geological Survey[3] (USGS), the Florida Department of Environmental Protection[4] (FDEP) and the National Oceanic & Atmospheric Administration[5] (NOAA). Additional archived weather data is available from the National Climate Data Center[6] (NCDC) which serves as a repository for the National Weather Service Cooperative (COOP) Station Network.

4.2.1   Water Level

Water level data used as boundary conditions and for model comparison come from the FDEP and USGS. Descriptions and locations of the available 1998 water level, wind speed and direction, precipitation and salinity measuring stations are shown in Table 4.1 and locations of these stations within the IRL estuary system are shown in Figure 4.1. The exact locations of the water level stations located within the IRL fine grid are shown in Appendix H.

Four connections exist between the IRL domain and the Atlantic Ocean (Figure 2.7). These four connections, Ponce de Leon Inlet, Sebastian Inlet, Ft. Pierce Inlet and St. Lucie Inlet, were chosen as open boundaries in both of the boundary-fitted grids. The grid cells representing the inlets are close enough to FDEP measured water level stations such that the corresponding measured water level data, with slight adjustment, can be used as open boundary conditions in the model (Figure 4.2).

4.2.2   Wind Speed and Direction

The surface wind boundary condition is supplied using hourly wind magnitude and direction data collected at five stations spanning the length of the lagoon.

---

[2] http://www.sfwmd.gov/

[3] http://www.usgs.gov/

[4] http://www.dep.state.fl.us/

[5] http://www.noaa.gov/

[6] http://www.ncdc.noaa.gov/

Table 4.1: Descriptions of the available 1998 IRL measured water level (ζ), wind speed and direction (W), precipitation (P) and bottom and surface salinity (S_b and S_s) data collected by the FDEP, SJRWMD and USGS. Values indicating the percentage of missing data are shown at stations where data were collected. The temporal spacing of all data is one hour except for water level which is six minutes.

Station Number	Station Name	Agency	Longitude (deg W)	Latitude (deg N)	UTM X(m)	UTM Y(m)	ζ (%)	W (%)	P (%)	S_s/S_b (%/%)
872-1147	Ponce Inlet South	FDEP	80 54 53	29 03 46	508307	3214992	0.00	0.00		
872-1164	NSB - Mosquito Lagoon	FDEP	80 55 06	29 01 22	507970	3210563	8.72			33.44/28.28
678	Playalinda	SJRWMD	80 49 27	28 55 40	517160	3200026			0.00	
643	Big Flounder	SJRWMD	80 51 16	28 45 20	514233	3180942			0.00	
02248380	Haulover Canal near MIMS	USGS	80 45 17	28 44 10	523957	3178825	6.91	25.14		37.66/30.72
872-1456	Titusville Brewer Causeway	FDEP	80 47 53	28 37 13	519750	3165979	2.86	0.00		13.63/24.67
313	Fox Lake	SJRWMD	80 52 19	28 35 16	512544	3162353			0.00	
670	Addison Creek	SJRWMD	80 48 47	28 32 00	518312	3156330			0.00	
682	Kiwanis Park - MI	SJRWMD	80 40 37	28 21 36	531680	3137156			0.00	
872-1647	Merritt Causeway (East)	FDEP	80 38 54	28 21 25	534473	3136842	0.00			2.73/ 9.73
872-1648	Merritt Causeway (West)	FDEP	80 43 09	28 21 20	527543	3136684	4.98			41.39/42.44
872-1789	Banana River - Carter's Cut	FDEP	80 36 20	28 08 56	538727	3113799	4.71	0.00		37.59/ 6.02
109	Melbourne Field Station	SJRWMD	80 40 17	28 08 30	532290	3112972			0.00	
872-1843	Melbourne Causeway	FDEP	80 35 31	28 04 59	540113	3106533	3.46			37.89/12.15
135	Crane Creek at City Hall	SJRWMD	80 40 05	28 04 22	532638	3105341		0.00		
410	Palm Bay	SJRWMD	80 35 51	28 01 33	539588	3100162		0.00		
126	Valkaria Airport	SJRWMD	80 33 37	27 54 44	543292	3087590		0.00		
122	Barefoot Bay	SJRWMD	80 32 06	27 53 20	545789	3085014		0.00		
872-2004	Sebastian Inlet	FDEP	80 26 55	27 51 36	554300	3081876	0.00			
2750170802955600	Sebastian River near RR BR	USGS	80 29 56	27 50 17	549367	3079397				30.72/5.84
215	S-157	SJRWMD	80 32 25	27 49 48	545294	3078489			0.00	
172	West Canal	SJRWMD	80 30 40	27 45 27	548199	3070469				
02251800	Indian River at Wabasso	USGS	80 25 39	27 45 14	556413	3070136	24.70		0.00	
686	Vero Beach	SJRWMD	80 24 27	27 39 42	558460	3059899				
872-2125	Vero Bridge	FDEP	80 22 18	27 37 54	562011	3056593	16.97			12.39/19.70
872-2208	Ft. Pierce Causeway	FDEP	80 19 29	27 28 16	566740	3038864	0.28			27.59/40.80
872-2213	Ft. Pierce Inlet	FDEP	80 17 57	27 28 05	569267	3038509	0.00		0.00	
872-2375	South Point St. Lucie Inlet	FDEP	80 09 55	27 09 51	582721	3004930	0.00			
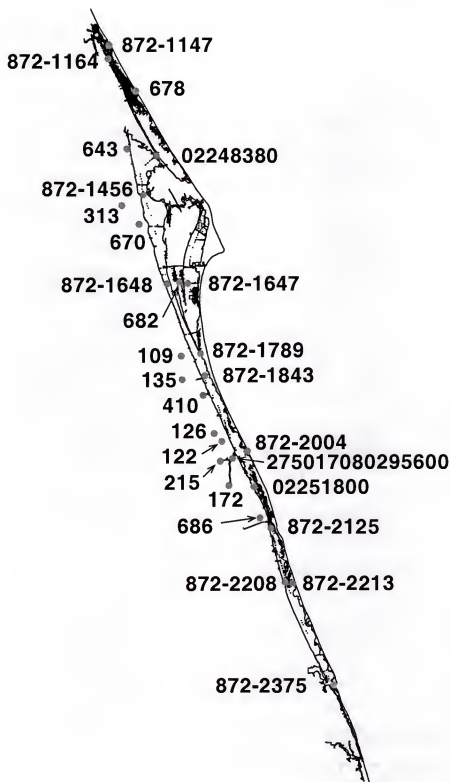
Figure 4.1: Locations of the available 1998 IRL measured water level, wind speed and direction, precipitation and salinity stations operated by the FDEP, SJRWMD and USGS.
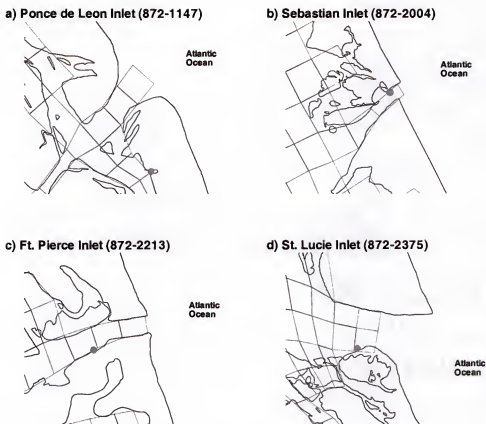
Figure 4.2: Measured water level stations near inlets.

Four of these stations were operated by the FDEP, Banana River (#872-1789), Ft. Pierce Inlet (#872-2213), Ponce Inlet (#872-1147) and Titusville Brewer Causeway (#872-1456), while the remaining station was operated by the USGS, Haulover Canal (#02248380).

The hourly wind magnitude and direction data were converted into x- (East-/West) and y-velocity (North/South) components for the period July 16, 1996 through September 30, 1999. The wind data were nearly complete for the entire time period except for the station located in Titusville (FDEP #872-1456) which ended on September 21, 1999 and the station in Haulover Canal (USGS #02248380) which ended September 30, 1998. Missing x- and y-velocity components are recorded as
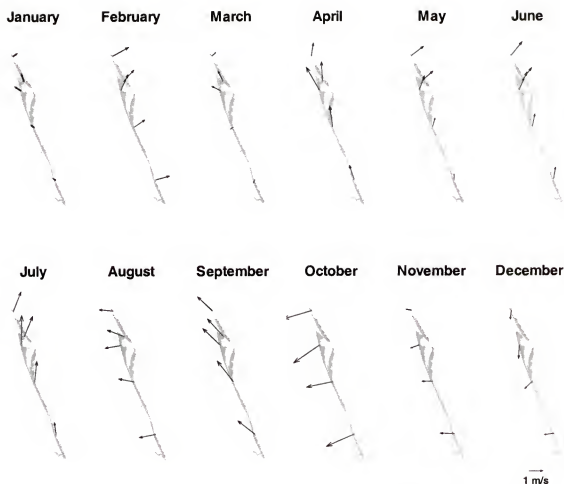
Figure 4.3: Monthly averaged wind at the 4 FDEP and 1 USGS stations.

values of -999.00 in the input files. These values signal the inverse distance interpolation scheme in CH3D to exclude the missing values when calculating the wind field in the model domain. Figure 4.3 shows monthly averaged wind speed and direction during 1998.

4.2.3 Tributary and Runoff Discharges

Discharge and runoff boundary conditions are imposed using the 15-minute, 30-minute, daily and monthly measured discharge data described in Table 4.2. Because no flow data were available for the Halifax River a value had to be approximated. The Halifax River watershed is 970×10^6 m^2 compared to 172×10^6 m^2

for Spruce Creek; thus, multiplying Spruce Creek flow by 6 ($\approx \frac{970}{172}$) is used as an approximation for the Halifax River Flow.

All of the measured discharge sites occurred to the south of Merritt Island (Figure 4.4). For the most part, the 1998 measured discharge data were continuous with the exception of small temporal gaps which were replaced by linear interpolation. Most of the northern tributaries and all of the runoff amounts were determined by other means, such as by using watershed fractions of other measured tributaries or by using simulated runoffs as calculated by watershed models. Descriptions of the actual 137 discharge and runoff sources used by the model and how they were obtained are shown in Table 4.3.

To visualize the quantity and location of flow into the IRL, the lagoon is broken into 9 segments (Du, 2000). Table 4.4 defines the 9 segments and Figure 4.5 plots the segments location within the IRL system. Using this segmentation scheme, the monthly and yearly totals of discharge into IRL during 1998 are shown in Figures 4.6 and 4.7, respectively.

4.2.4 Non-tributary Discharge

In addition to year-long tributary discharges, the USGS also measured discharge through Haulover Canal (#02248380) during 1998. The location of the discharge site is the same as that of the water level, wind, and salinity station shown in Table 4.1.

Four times during 1998 (January 15, February 25, May 27, and July 9), the USGS used ADCP to measure flow through the Ponce de Leon, Sebastian and Ft. Pierce Inlets and twice during 1998 (January 15 and May 27) measured flow through St. Lucie Inlet.

When analyzing the ADCP inlet flow data at Ponce de Leon Inlet for the January 15 sampling date, it became apparent that there was a problem with the times supplied with the discharge data. All ADCP data set comparisons between

Table 4.2: Descriptions of the available 1998 IRL measured discharge data.

Station Number	Station Name	Agency	Longitude (deg W)	Latitude (deg N)	UTM X(m)	UTM Y(m)	Frequency
02249007	Eau Gallie River at HG	USGS	80 38 49	28 07 35	534695	3111286	15 min
02249518	Crane Creek at US 1	USGS	80 36 08	28 04 38	539094	3105838	daily
02249515	Hickory St. Drainage Canal	USGS	80 36 48	28 04 17	538015	3105203	daily
02249510	Crane Creek at Babcock St.	USGS	80 37 17	28 04 06	537225	3104862	daily
02250030	Turkey Creek at Palm Bay	USGS	80 35 46	28 01 00	539728	3099147	30 min
02251500	N. Prong Seb River near Micco	USGS	80 31 28	27 51 21	546843	3081357	15 min
210	C-54 Canal at S-157	SJRWMD	80 31 28	27 49 51	545321	3078582	hourly
02251767	Fellsmere Main Canal	USGS	80 32 04	27 49 49	545869	3078522	30 min
02251000	S. Prong Seb. River near SR 512	USGS	80 30 22	27 46 09	548686	3071764	15 min
02252500	North Canal near Vero Beach	USGS	80 25 00	27 41 32	557540	3063279	15 min
02253000	Main Canal at Vero Beach	USGS	80 24 10	27 38 54	558933	3058424	15 min
02253500	South Canal near Vero Beach	USGS	80 23 24	27 36 11	560218	3053415	15 min
S99.S	S-99 Spillway on Canal C-25	SFWMD	80 28 18	27 28 14	552216	3038705	daily
S49.S	S-49 Spillway on Canal C-24	SFWMD	80 21 32	27 15 42	563478	3015621	daily
S97.S	S-97 Spillway on Canal C-23	SFWMD	80 20 26	27 12 19	565326	3009985	daily
S80.S	S-80 Spillway on St. Lucie Canal	SFWMD	80 17 05	27 06 40	570915	2998986	daily

Table 4.3: Descriptions of the 137 sources of discharge data used in the model. The data are grouped by time interval: 15 minutes, 30 minutes, daily, and monthly. N1 (north), E1 (east), E2, E3, E4, W1 (west), W2, W3, W4 are the SJRWMD simulated ungauged discharges for Sebastian River. IR1 (Indian River), IR2, IR3, BR1 (Banana River), BR2, BR3, BR4, BC (Barge Canal), SC1 (Sykes Creek), NH1 (Newfound Harbor), PI1 (Pine Island) are SJRWMD simulated ungauged discharges for Merritt Island south of SR 405.

Location	Data
Eau Gallie River	Observed by USGS
Sebastian River (North)	Fellsmere Canal (observed by USGS)
	+ C54 @ S157 (collected by SJRWMD)
	+ North Prong Sebastian (observed by USGS)
	+ N1
Sebastian River (South)	South Prong Sebastian (observed by USGS)
	+ E1 + E2 + E3 + E4
	+ W1 + W2 + W3 + W4
North Canal	Observed by USGS
Main Canal	Observed by USGS
South Canal	Observed by USGS
Turkey Creek	Observed by USGS
Halifax River	6 x Spruce Creek
Spruce Creek	Simulated by SJRWMD
Turnbull Creek	Simulated by SJRWMD
Merritt Island	5/17 x PI1
Merritt Island	12/17 x PI1
Merritt Island	BR4
Merritt Island	IR3 + 1/3 x BC1
Merritt Island	BR3 + 1/3 x BC1
Merritt Island	IR2
Merritt Island	BR1 +BR2
Merritt Island	IR1
Merritt Island	SC1 + NH1 + 1/3 x BC1
Crane Creek at US1	Simulated by USGS
Goat Creek	Simulated by SJRWMD
Trout Creek	Simulated by SJRWMD
St. Lucie (North)	S49 + S97 + S99 (measured by SFWMD)
St. Lucie (South)	S80 (measured by SFWMD)
All other basins (113 sub-basins)	Simulated by SJRWMD
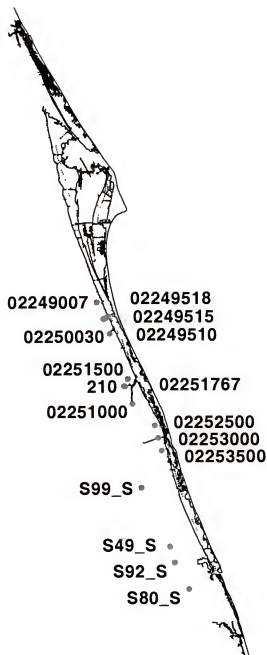
Figure 4.4: Locations of the available 1998 IRL measured discharge stations belonging to the SFWMD, SJRWMD and USGS.

simulated flow rate and measured flow rate looked good except for the January 15 data set at Ponce de Leon Inlet. For this data set, the simulated results appeared to be shifted by one hour. When this data set was further analyzed, it was noticed that the first ADCP measurement at Ponce de Leon Inlet was taken an hour before

Table 4.4: The boundaries of each of the nine segments of Indian River Lagoon. All segments have the shore as the western and eastern boundaries.

Segment Number	North Boundary	South Boundary	Tidal Inlets	Major Tributaries
1	Shore	Shore	Ponce de Leon	Halifax River Spruce Creek Haulover Canal
2	Shore	Bennett Causeway (S. R. 528)		Turnbull Creek Canaveral Canal
3	Bennett Causeway	Dragon Point (S. R. 3)		Sykes Creek
4	Shore	NASA Causeway (S. R. 405)		Haulover Canal Turnbull Creek Big Flounder Creek
5	NASA Causeway (S. R. 405)	Melbourne Causeway (S. R. 500)		Canaveral Canal Horse Creek Eau Gallie River
6	Melbourne Causeway	6-7 free connection		Crane Creek Turkey Creek Goat Creek Trout Creek
7	7-6 free connection	Wabasso Causeway (S. R. 510)	Sebastian	Fellsmere Canal Sebastian River
8	Wabasso Causeway (S. R. 510)	8-9 free connection		North Canal Main Canal South Canal
9	9-8 free connection	Shore	Ft. Pierce St. Lucie	St. Lucie River

sunrise. Table 4.5 shows the difference between the time of first ADCP measurement and apparent sunrise for the four ADCP inlet measuring date at the four inlets. The time of apparent sunrise is calculated using the NOAA Surface Radiation Research Branch Sunrise/Sunset Calculator[7]. The latitudes and longitudes used to calculate the sunrise at the four inlets, Ponce de Leon, Sebastian, Ft. Pierce and St. Lucie Inlets, are those of the FDEP inlet stations: #872-1147, #872-2004, #872-2213 and #872-2375, respectively.

The January 15 ADCP measurement at Ponce de Leon Inlet is the only set of data collected before sunrise. Assuming that data collection does not occur before

---

[7]http://www.srrb.noaa.gov/highlights/sunrise/sunrise.html
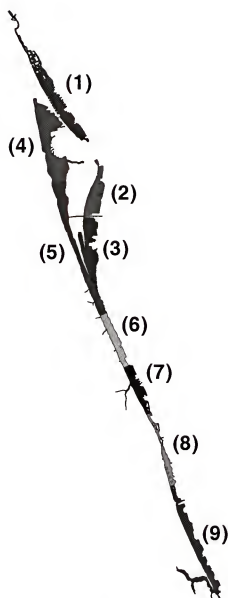
Figure 4.5: Indian River Lagoon 9 segment scheme.

sunrise because of the dangerous nature of sampling in an actively traversed inlet, this set of data is off by at least 58 minutes; hence, an error in the times for this data set is assumed and the times of the measured data for this data set were increased by one hour. New time series comparisons between the adjusted January 15 flow rate data and simulated flow rate data now show good agreement. Any flow rate
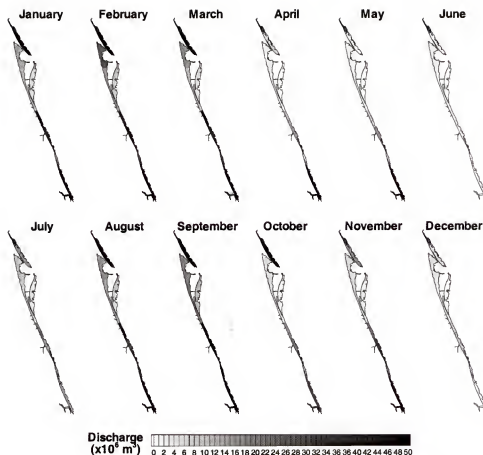
Figure 4.6: Monthly discharge into the 9 segments of the IRL.

comparisons from here on, unless otherwise stated, will using the adjusted January 15 flow rate data at Ponce de Leon Inlet.

4.2.5 Precipitation

Hourly precipitation data were recorded at SJRWMD rain gauges at the following 13 locations along the length of the lagoon: Addison Creek (670), Barefoot Bay (122), Crane Creek at City Hall (135), Big Flounder (643), Fox Lake (313), Kiwanis Park (682), Melbourne (109) Palm Bay (410), S157 (215), Valkaria Airport (126), Playalinda at Turtle Mound (678), Vero Beach (686), and West Canal (172). The hourly precipitation data were continuous from January 1, 1998 to September 30, 1999.
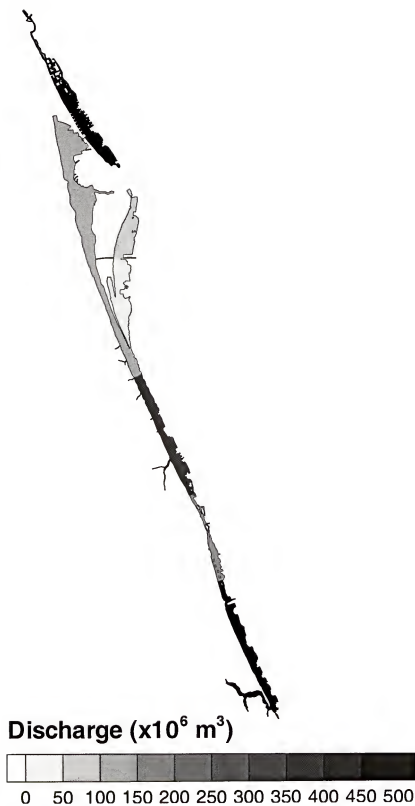
Figure 4.7: Total yearly discharge into the 9 segments of the IRL.

Table 4.5: A comparison between the time of collection and the time of apparent sunrise on the four days the USGS collected flow rate measurements in Ponce de Leon, Sebastian, Ft. Pierce and St. Lucie Inlet during 1998. All times shown are in EST.

Date	Ponce de Leon Inlet	Sebastian Inlet	Ft. Pierce Inlet	St. Lucie Inlet
	Time of apparent sunrise			
January 15 (15)	7:18am	7:14am	7:13am	7:12am
February 25 (56)	6:53am	6:50am	6:50am	6:49am
May 27 (147)	5:27am	5:27am	5:27am	5:27am
July 9 (190)	5:32am	5:33am	5:33am	5:33am
	Time of first ADCP measurement			
January 15 (15)	6:20am	7:16am	7:41am	7:39am
February 25 (56)	7:53am	7:03am	8:05am	
May 27 (147)	6:00am	5:56am	5:59am	
July 9 (190)	7:09am	6:03am	5:36am	5:50am
	Time difference between first ADCP measurement and apparent sunrise			
January 15 (15)	-0:58	+0:02	+0:28	+0:27
February 25 (56)	+0:60	+0:13	+0:75	
May 27 (147)	+0:33	+0:29	+0:32	
July 9 (190)	+0:97	+0:30	+0:03	+0:23

Precipitation data collected by NOAA were also available. Daily rainfall totals for Titusville (8942), Melbourne (5612) and Vero Beach (9219) were available from January 1, 1966 through December 31, 1993. In addition, daily rainfall totals for Daytona Beach (2158), Titusville (8942), Ft. Pierce (3207), Melbourne (5612), and Vero Beach (9219) were available from January 1, 1998 through December 31, 1998. However, because of the extensiveness of the SJRWMD hourly data, the NOAA daily precipitation data was not used.

4.2.6   Evaporation

The only source of evaporation data in the IRL domain was the NOAA daily pan evaporation data measured at Vero Beach from January 1, 1966 through December 31, 1997 and November 1, 1998 through September 30, 1999. The missing period during 1998 posed a major problem because 1998 was the designated simulation period. Four methods were explored to estimate the missing 1998 evaporation data:

- 1) Use historical monthly evaporation averages at the Vero Beach evaporation station applied spatially constant over the entire lagoon.

- 2) Calculate the monthly historical percentile rainfall totals at the Vero Beach (9219) rainfall gauge. Assuming that evaporation varies inversely with precipitation, calculate the inverse of the precipitation percentile. Finally, calculate the evaporation at the Vero Beach station using the monthly percentile inverse and the historical evaporation at the Vero Beach. The resulting monthly historical percentile based evaporation (Table 4.6) is then applied spatially constant over the entire lagoon.

- 3) Use the same process as Method 2, except the precipitation percentiles and resulting evaporation is also calculated using the NOAA Titusville (8942) (Table 4.7) and Melbourne (5612) (Table 4.8) rain gauges. The resulting monthly historical percentile based evaporation is then spatially interpolated over the entire lagoon.

- 4) Use the evaporation measured at the Belle Glade Experiment Station (COOP ID #080611) applied spatially constant over the entire lagoon. The Belle Glade Experiment Station is located near, but outside of, the IRL domain ($26.650°$ N / $80.633°$ W) approximately 70 km southwest of St. Lucie Inlet.

Table 4.6: Estimated monthly evaporation at the NOAA Vero Beach Station (9219).

Month	Precipitation			Evaporation	
	1998	Historical	Percentile	Vero Beach	Calculated
	(cm)	(cm)	(%)	(cm)	(cm)
January	11.26	6.93	83.9	8.09	6.94
February	17.69	8.09	98.2	10.25	7.56
March	18.32	9.18	94.6	14.51	12.44
April	3.84	5.91	37.5	17.18	17.80
May	10.01	12.01	44.6	19.16	19.04
June	2.49	16.46	5.4	18.64	26.48
July	20.08	16.34	66.1	18.27	17.24
August	24.23	17.07	91.1	16.99	14.24
September	20.91	18.23	76.8	15.23	13.77
October	8.71	14.33	41.1	13.22	13.81
November	13.04	9.22	87.0	9.49	8.05
December	2.34	5.15	19.6	7.69	8.96

Table 4.7: Estimated monthly evaporation at the NOAA Titusville Station (8942).

Month	Precipitation			Evaporation	
	1998	Historical	Percentile	Vero Beach	Calculated
	(cm)	(cm)	(%)	(cm)	(cm)
January	7.02	6.03	66.0	8.09	7.56
February	21.68	8.14	98.1	10.25	7.61
March	10.26	8.44	68.5	14.51	13.91
April	0.58	6.58	9.6	17.18	19.42
May	2.75	9.38	10.0	19.16	23.89
June	2.07	17.47	5.8	18.64	24.86
July	12.66	18.94	21.2	18.27	19.88
August	18.00	18.60	40.4	16.99	17.15
September	22.12	17.65	82.7	15.23	13.18
October	2.85	12.50	10.0	13.22	15.74
November	4.98	8.28	35.4	9.49	9.62
December	5.06	5.51	57.4	7.69	7.24

A comparison between the evaporation estimated at the NOAA Vero Beach Station using Methods 1, 2, and 4 is shown in Figure 4.8.

Table 4.8: Estimated monthly evaporation at the NOAA Melbourne Station (5612).

| Month | Precipitation | | | Evaporation | |
	1998 (cm)	Historical (cm)	Percentile (%)	Vero Beach (cm)	Calculated (cm)
January	13.73	6.17	94.2	8.09	6.53
February	15.60	6.90	98.1	10.25	7.61
March	12.45	7.80	80.4	14.51	13.07
April	2.14	4.65	42.6	17.18	17.62
May	2.16	10.61	6.0	19.16	25.66
June	0.41	14.36	5.8	18.64	24.86
July	23.15	13.38	94.6	18.27	15.73
August	20.44	12.53	94.2	16.99	14.15
September	26.31	17.03	90.4	15.23	12.80
October	3.32	10.58	16.7	13.22	15.28
November	14.05	7.56	90.4	9.49	7.54
December	6.49	5.31	71.2	7.69	6.84

To convert from pan evaporation to lake evaporation a coefficient of 0.78 was used. Several 1998 test simulations using pan coefficients of 0.66 and 0.90 were conducted; however, error analysis performed on the test simulations did not show marked improvement of the coefficient value of 0.78. Yearly and monthly plots of precipitation, evaporation and combined precipitation and evaporation for the different evaporation methods appear in Appendix I.

4.2.7   Salinity

As was mentioned earlier, Table 4.1 lists some of the available measured salinity data during 1998 within the IRL (Figure 4.1). These salinity data were collected hourly during 1998; however, much of the data is either bad or missing. The exact locations of the salinity stations located within the IRL fine grid are shown in Appendix H.

The IRL coarse and fine grids were developed to study large scale hydrodynamics and transport within the lagoon; hence, localized small scale flow and salinity patterns may not be accurately resolved. For example, the measured salinity at
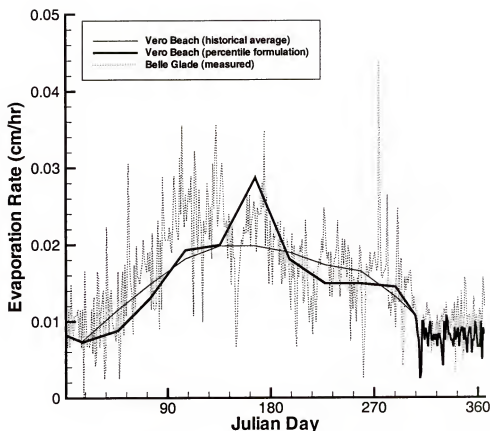
Figure 4.8: A comparison between the historical monthly Vero Beach evaporation, the historical monthly percentile evaporation at Vero Beach and the measured evaporation at the Belle Glade Experiment Station (COOP ID #080611).

Sebastian River Station (USGS #275017080295600) represents a small scale local feature which is difficult to simulate. Salinity boundary data are uncertain and the station is located far enough upstream such that it is close to the boundary. Thus, relatively high RMS errors occur when the measured salinity is compared with the simulated salinity especially in the lower water column. Smaller scale studies which use smaller, denser grid systems focused on the region near the river are better able to simulate the often fluctuating salinity, for example the Sebastian River study of Sucsy et al. (1998) and the Turkey Creek studies of Sucsy and Morris (1998); Moustafa and Hamrick (1994). Thus, to keep the error associated with the Sebastian River Station from affecting the lagoon-wide average error analysis, average comparisons between

simulated and measured salinity are performed both with and without the Sebastian River Station.

Additional bi-weekly "snapshot" salinity data were measured as part of the Indian River Lagoon Water Quality Monitoring Network (IRL-WQMN) (Steward and Higman, 1991; SJRWMD et al., 1993; Sigua et al., 1996). The IRL-WQMN was established in 1988 as a coordinated multi-agency project spanning the entire length of the IRL system (Figure 4.9). Water quality monitoring in the IRL system consisted of sampling at regular intervals (monthly) for a suite of parameters agreed upon by the various participating agencies: SJRWMD, SFWMD, Volusia County, Brevard County, Indian River County, and NASA-Dynamac.

The IRL-WQMN salinity data are less useful because they are not continuous; however, the data are collected at many more locations than the continuous salinity measurements mentioned previously. The locations of the IRL-WQMN sampling stations are shown in Table 4.9.

There were no long-term salinity measurements available at the inlets where the open boundaries were placed; thus, a constant oceanic salinity value of 35 ppt was applied to water flowing into the lagoon at the oceanic boundaries. There were also no long-term salinity measurements at the numerous freshwater discharge sources. A constant concentration of 0 ppt was applied to all incoming discharges with one exception, the discharge from Halifax River. Because the northern boundary of the grid systems (Figures 2.7 and 2.8) does not fully extend up the Halifax River, significant mixing will have occurred by the time the discharge reaches the northern boundary of the model. Thus, a 20 ppt concentration was used for the Halifax River discharge.

4.3 Sensitivity and Calibration Simulations

4.3.1 Water Level Used as Tidal Forcing

When measured inlet water level is used as a tidal boundary condition, it is necessary to fill temporal gaps in the collected data. While three of the four FDEP
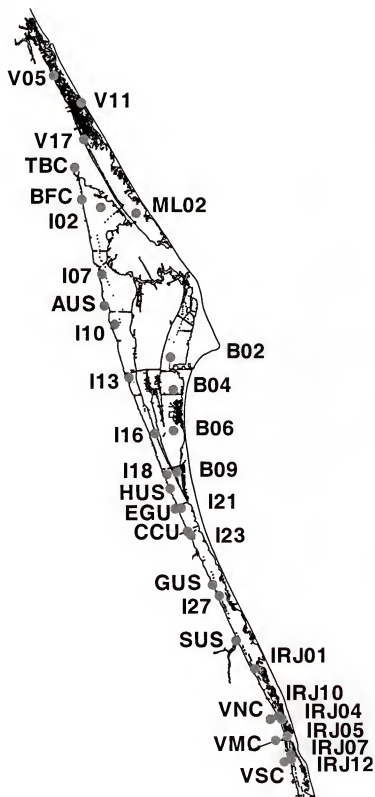
Figure 4.9: IRL-WQMN water sampling stations which have measured salinity during 1998.

Table 4.9: Locations of the 1998 IRL-WQMN stations which contain measured salinity data.

Station ID	Longitude (W)	Latitude (N)	UTM (X)	UTM (Y)
B02	80 38 22	28 26 01	535330	3145321
B04	80 38 00	28 22 00	535951	3137907
B06	80 38 00	28 17 00	535979	3128675
B09	80 37 32	28 11 56	536771	3119323
BFC	80 50 42	28 45 18	515155	3180882
CCU	80 36 08	28 04 39	539105	3105884
EGU	80 37 50	28 07 25	536306	3110983
GUS	80 32 41	27 58 05	544800	3093780
HUS	80 38 31	28 09 55	535173	3115595
I02	80 48 02	28 44 20	519497	3179103
I07	80 47 54	28 36 12	519739	3164086
I10	80 46 08	28 30 04	522640	3152768
I13	80 44 10	28 23 34	525874	3140773
I16	80 40 36	28 16 40	531731	3128048
I18	80 38 56	28 11 40	534482	3118824
I21	80 37 00	28 07 30	537669	3111141
I23	80 35 40	28 04 12	539872	3105056
I27	80 31 46	27 56 44	546312	3091293
IRJ01	80 26 56	27 47 48	554311	3074834
IRJ04	80 23 14	27 41 33	560443	3063324
IRJ05	80 22 32	27 39 28	561613	3059484
IRJ07	80 22 04	27 37 11	562402	3055272
IRJ10	80 23 39	27 41 57	559755	3064059
IRJ12	80 22 01	27 36 34	562490	3054134
ML02	80 43 05	28 43 35	527556	3177735
SUS	80 29 29	27 51 15	550098	3081185
TBC	80 51 41	28 49 14	513546	3188142
V05	80 54 34	29 00 29	508841	3208910
V11	80 50 41	28 57 09	515153	3202762
V17	80 50 22	28 52 41	515679	3194515
VNC	80 24 49	27 41 34	557841	3063342
VMC	80 24 08	27 38 57	558988	3058517
VSC	80 22 58	27 36 17	560930	3053603

-

inlet stations required little gap filling for the 1998 simulations, the St. Lucie Station required the whole year to be regenerated (Table 4.10).

Table 4.10: Missing 1998 water level data at the FDEP inlet stations used as tidal boundary conditions.

Station Name	Station Number	Missing Period
Ponce Inlet South	872-1147	Nov. 11 - Dec. 1
Sebastian Inlet	872-2004	Apr. 23 - May 3
		Dec. 17 - Dec. 31
Ft. Pierce Inlet	872-2213	Sep. 14 - Oct. 1
South Point St. Lucie Inlet	872-2375	Jan. 1 - Dec. 31

The water level gaps in the measured data collected at the FDEP inlet stations were reconstructed by the SJRWMD using the following technique (Sucsy, 2000):

- Perform a tidal prediction at the site using previously calculated short-period tidal harmonics (diurnal, semi-diurnal periods and higher harmonics only).

- Tidally filter a nearby station that has good data for the time period of the missing data using a 30-hour filter to remove short-period tidal energy. (The station used to fill another station was previously determined by correlating filtered water level data for all stations. The station used to fill is the one with the highest correlation.)

- Add the filtered water level data from the neighboring station to the tidal prediction at the station to be filled resulting in a reconstructed hourly water level series.

- Disaggregate the series into 6-minute values using a spline-fit to interpolate from the hourly data to 6-minute intervals to correspond to the time interval of the other measured FDEP water level data.

The FDEP water level stations used as tidal forcing at Ponce de Leon and Ft. Pierce Inlets are not located in the outermost grid cell at which the tidal elevation is forced (Figure 4.2). If the simulated water level at the cell containing the actual

location of the FDEP water level station is compared to the measured water level, a slight phase shift can be seen. Hence, the measured water levels near Ponce de Leon and Ft. Pierce Inlets were given a negative phase shift before being used as boundary conditions at the inlets. This phase shift was determined by phase shifting the measured water level by varying amounts and performing a 30-day simulation using the phase shifted water levels as forcing. The simulated and measured water levels were then compared and root mean square (RMS) errors between the simulated and measured water level at the Ponce de Leon and Ft. Pierce Inlet Stations were calculated using

$$E_{rms} = \sqrt{\sum_{n=1}^{N} \frac{\left(X_{measured}(n) - X_{simulated}(n)\right)^2}{N}} \tag{4.1}$$

where $X(n)$ is the nth measured or simulated value, and N is the total number of observed and measured values.

The phase shifted forcing which results in the smallest RMS error when comparing water level at Ponce and Ft. Pierce Inlets are then used in all subsequent simulations. These phase shifts, at Ponce de Leon and Ft. Pierce Inlets, were determined to be -290s and -120s, respectively. Using the phase shifted water level as inlet forcing, the RMS errors at the Ponce de Leon and Ft. Pierce Inlets Stations decreased 56% (to 0.68 cm) and 28% (to 0.42 cm), respectively, for the 30-day test simulation.

Another effect of applying water level data measured inside the throat of an inlet to the throat is that the measured water level inside the lagoon will over-estimate the mean sea level (MSL). In general, the mean water level (MWL) inside of an inlet is higher than the MSL of the open ocean. This difference, $MWL - MSL$ is know as superelevation, δ, and can be caused by such factors as morphology, tide, runoff, salinity, wind, waves, etc[8]

---

[8]Further discussion of superelevation can be found in Mehta (1990) and Mann and Mehta (1993).

During 1998, the mean measured water level at the four FDEP inlet stations, #872-1147, #872-2004, #872-2213 and #872-2375), relative to NAVD88, was -24.3 cm, -36.6 cm, -36.2 cm, and -36.2 cm, respectively. Three of the values being -36 cm implies that the superelevation at the measured Ponce de Leon Inlet station is approximately 12 cm. However, because of the possibility of a South to North setup of the Atlantic Ocean along Florida's east coast, it would not be prudent to lower the mean water level by that amount without further study.

To more accurately determine the amount of superelevation inside Ponce de Leon Inlet, it is necessary to calculate the MWL at several stations both inside and outside of the inlet. Unfortunately, during the 1998 simulation period, there were not enough water level stations to accurately determine the superelevation inside the inlet. However, during July and August 1996, an array of water level measurements were taken by the U. S. Army Engineer Waterways Experiment Station (WES) in the vicinity of Ponce de Leon Inlet (King Jr. et al., 1999). Descriptions of the WES stations as well as other available FDEP and NOAA water level stations near Ponce de Leon Inlet and along the east coast of Florida where water level data were measured during July and August 1996 are shown in Table 4.11.

To determine the superelevation inside of Ponce de Leon inlet, the water level stations must be leveled accurately. The SPRSBAY stations were leveled to NGVD29 and converted, for the purposes of this study, to NAVD88 using Vertcon[9], a vertical datum conversion utility developed by the National Geodetic Survey (NGS). The DWG stations were not surveyed directly but were converted to NGVD29 using long-term average values as described in Howell (1997). These stations were also converted to NAVD88 via the Vertcon program. The remaining FDEP and NOAA water level data were supplied in NAVD88. A plot of MWL and superelevation calculated with respect to DWG1IN1 (Figure 4.10) shows the that during July and August 1996

---

[9]http://www.ngs.noaa.gov/TOOLS/Vertcon/vertcon.html

Table 4.11: Descriptions of the available water level data near Ponce de Leon Inlet and associated offshore stations during July and August 1996. "Direct" refers to direct leveling in NAVD88, "Converted" refers measurement in NGVD29 with a conversion to NAVD88 and "Approx." refers to approximation to NGVD29 followed by a conversion to NAVD88.

Station Number	Station Name	Agency	Leveling Method	Longitude (deg W)	Latitude (deg N)	UTM X(m)	UTM Y(m)
SPRSBAY1	River North (Deepwater Marina)	WES	Converted	80 56 26	29 05 27	505809	3218085
DWG1INT1	Offshore (North of Inlet)	WES	Approx.	80 54 37	29 05 26	508750	3218054
DWG1OTH1	Inlet Throat (Outer Inlet Throat)	WES	Approx.	80 54 29	29 04 36	508962	3216521
DWG1EBBI	Ebb Shoal	WES	Approx.	80 53 54	29 04 34	509914	3216448
SPRSBAY2	River West (Utility Company)	WES	Converted	80 56 20	29 03 57	505973	3215299
872-1147	Ponce Inlet South	FDEP	Direct	80 54 53	29 03 46	508307	3214992
SPRSBAY3	River South (Riverview Restaurant)	WES	Converted	80 54 18	29 02 16	509273	3212203
872-1164	NSB - Mosquito Lagoon	FDEP	Direct	80 55 06	29 01 22	507970	3210563
872-1604	Trident Pier - Port Canaveral	NOAA	Direct	80 35 36	28 24 54	539853	3143274
872-2004	Sebastian Inlet	FDEP	Direct	80 26 55	27 51 36	554300	3081876
872-2213	Ft. Pierce Inlet	FDEP	Direct	80 17 57	27 28 05	569267	3038509
872-2375	South Point St. Lucie Inlet	FDEP	Direct	80 09 55	27 09 51	582721	3004930

little setup was present off the east coast of Florida and that the difference in MWL between the FDEP station at the Ponce de Leon Inlet and FDEP stations at the other inlets is most likely caused by the superelevation effect at Ponce de Leon Inlet. The amount of superelevation at the boundary cell of the boundary-fitted grid is in the range of 8.1 to 12.9 cm.
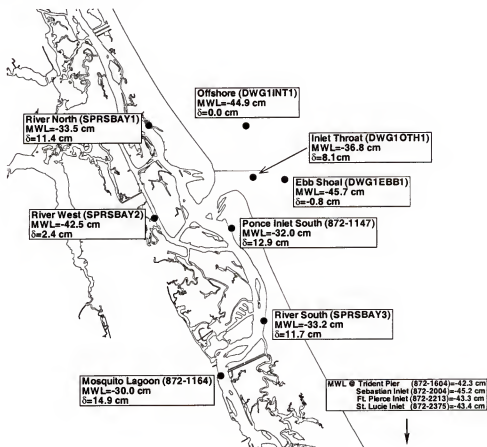
Figure 4.10: Mean water levels (MWLs) at several stations near Ponce de Leon Inlet during July and August 1996. The superelevation, δ, is calculated with respect to the Offshore Station (DWG1INT1). All MWLs are calculated with respect to NAVD88.

Along with trying to determine the superelevation by means of measured data, simulations were performed which reduced the MWL of the boundary condition at Ponce de Leon Inlet by various amounts (Table 4.12). Without reduction of the MWL, the simulated flow through Haulover Canal is much stronger, -33 m^3/s, than

the measured value, 0.25 m^3/s, which forces the highly saline water in the Mosquito Lagoon into the northern IRL causing the RMS errors for the salinity comparisons to increase.

Table 4.12: The effect of reducing the MWL of the tidal forcing at Ponce de Leon Inlet on the accuracy of simulated water level, salinity and flow through Haulover Canal. Values shown are average RMS errors for all available stations. Values shown in parenthesis for water level exclude the Ponce de Leon Station (FDEP #872-1147) and Mosquito Lagoon Station (FDEP #872-1164) while values shown in parenthesis for salinity exclude the Sebastian River Station (USGS #275017080295600). Mean flow values are shown in parenthesis for flow through Haulover Canal and can be compared to the measured mean of 0.25 m^3/s with a positive value indicating flow into Mosquito Lagoon.

MWL Reduction (cm)	Water Level (cm)	Salinity (ppt)	Flow though Haulover Canal (m^3/s, m/s)
None	2.80(2.81)	7.63(7.40)	69.51(-33.26)
4	3.48(3.06)	6.15(5.77)	63.57(-20.04)
7	4.04(3.26)	4.83(4.33)	61.10(-10.06)
10	4.60(3.45)	3.53(2.92)	59.68(-1.09)
10.5	4.70(3.48)	3.40(2.79)	59.49(0.37)
11	4.80(3.52)	3.36(2.74)	59.34(1.82)
12	5.01(3.61)	3.39(2.79)	59.19(4.75)
13	5.23(3.71)	3.60(3.02)	59.25(7.75)

The RMS errors show that when the MWL of the Ponce de Leon Inlet Station is reduced by 11 cm and then used as tidal forcing, a minimum RMS error for salinity is achieved while the minimum RMS for flow rate through Haulover Canal is achieved when a 10.5 cm reduction of MWL is used (Figure 4.11). Thus, if the measured water level at the FDEP Ponce de Leon Inlet station is to be used as forcing, its MWL should be reduced by 10.5-11 cm to minimize simulation errors. Because of the possibility of errors due to poor evaporation or fresh water discharge, achieving a smaller error in flow rate through Haulover Canal was deemed more important than a slightly smaller average RMS error for salinity. Thus, all simulations performed from here on, will have as a boundary condition at Ponce de Leon Inlet, the water level at the FDEP

Ponce de Leon Inlet Station phase shifted by -290s and its MWL reduced by 10.5 cm.
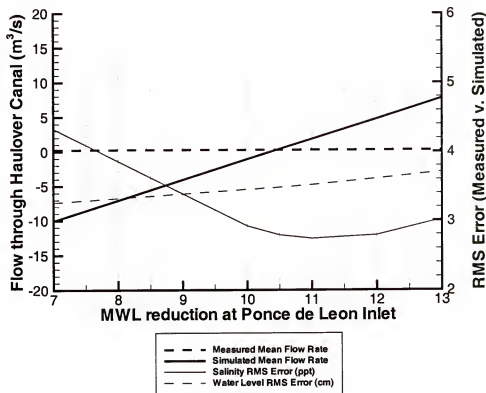
Figure 4.11: A comparison between simulated results and measured data showing the effect of reducing the mean water level (MWL) at Ponce de Leon Inlet. Positive flow is directed into Mosquito Lagoon.

When the MWL at the Ponce de Leon Inlet boundary is reduced so are the simulated MWL at the grid locations of all the measured water level stations all the way south to Sebastian Inlet. While this increases the simulated errors substantially at the two stations closest to Ponce de Leon Inlet: Ponce Inlet South (FDEP #872-1147) and Mosquito Lagoon (FDEP #872-1164), most simulated MWL are minimally affected and nearly all of the salinity RMS comparisons are improved. Thus, to keep the error associated with adjusting the MWL at Ponce Inlet from affecting the lagoon-wide average error analysis, average comparisons between simulated and measured

water level are performed both with and without the Ponce Inlet South and Mosquito Lagoon water level stations.

As discussed in Militello and Zarillo (2000), the Coronado Beach Bridge (just south of Ponce de Leon Inlet) alone accounts for 55% reduction of M_2 amplitude. Bridges and other sub-grid scale features such as the numerous islands in Mosquito Lagoon are not adequately resolved in the current fine grid. These features have the effect of, through friction, blocking the flow and causing the water to pile up between the north Mosquito Lagoon and Ponce de Leon Inlet. Since the current boundary-fitted grid cannot resolve these sub-grid scale features, the water does not pile up and becomes artificially low with respect to the measured data. To artificially simulate the friction, both lateral and vertical, induced by these sub-grid scale features, it is possible to make horizontal diffusion and bottom friction artificially large; however, because the width of the grid cells is too large in this region of many islands, the flow still passes through. Because of the relatively large number of computational cells in the fine grid of the entire IRL, making a "finer grid" to resolve the numerous islands would require significantly more simulation time. One possible method to solve this problem is to add a porosity term to the governing equations in a manner similar to Wang and Kim (2000). This, however, would require significant effort beyond the current scope of work.

4.3.2 Number of Tributary and Runoff Discharges

As described in Table 4.3, 137 available sources of fresh water discharge data are available for model simulations. The first 24 are the primary sources of fresh water discharge into the lagoon which include the simulated ungauged discharges into Sebastian River and the simulated ungauged discharge and runoff from Merritt Island. The remaining 113 simulated sources of discharge represent all remaining watersheds which were not either simulated or measured previously.

When simulating the transport of a conservative species (such as salinity or sediments) within the lagoon, boundary conditions must be specified at each of the fresh water discharge sources. If 137 sources are to be used, the task of specifying these boundary values becomes difficult because of a lack of spatially and temporally dense measured data. To determine if all 137 fresh water discharge sources are necessary for accurate simulation of hydrodynamics and salinity transport, two 1998 simulations were performed. The first used only the first 24 sources of fresh water discharge and the second used all 137 sources of fresh water discharge. An error analysis performed on the simulated water level, salinity and transport through Haulover Canal did not show any improvement in model accuracy when the 137 sources of fresh water discharge were used (Table 4.13); hence, from here on, all simulations use only 24 sources of fresh water discharge.

Table 4.13: The effect of using either 24 or 137 sources of fresh water discharge on the accuracy of simulated water level, salinity and flow through Haulover Canal. Values shown are average RMS errors for all available stations. Values shown in parenthesis for water level exclude the Ponce de Leon Station (FDEP #872-1147) and Mosquito Lagoon Station (FDEP #872-1164) while values shown in parenthesis for salinity exclude the Sebastian River Station (USGS #275017080295600). Mean flow values are shown in parenthesis for flow through Haulover Canal and can be compared to the measured mean of 0.25 m³/s with a positive value indicating flow into Mosquito Lagoon.

Number of Discharge Sources	Water Level (cm)	Salinity (ppt)	Flow though Haulover Canal (m³/s, m/s)
24	4.70(3.48)	3.40(2.79)	59.49(0.37)
137	4.51(3.27)	3.96(3.42)	59.68(1.74)

4.3.3 Evaporation Estimation Method

To determine which of the previously discussed evaporation estimation methods performs the best, a 1998 simulation was conducted using each method (Table 4.14). Each method simulated water level and flow rate through Haulover Canal

equally well; however, the spatial percentile formulation had the smallest RMS error between the simulated and measured salinity; thus, from here on, the spatial percentile formulation for estimating evaporation is used in all simulations.

Table 4.14: The effect of the four methods of estimating evaporation on the accuracy of simulated water level, salinity and flow through Haulover Canal. Values shown are average RMS errors for all available stations. Values shown in parenthesis for water level exclude the Ponce de Leon Station (FDEP #872-1147) and Mosquito Lagoon Station (FDEP #872-1164) while values shown in parenthesis for salinity exclude the Sebastian River Station (USGS #275017080295600). Mean flow values are shown in parenthesis for flow through Haulover Canal and can be compared to the measured mean of 0.25 m³/s with a positive value indicating flow into Mosquito Lagoon.

Evaporation Estimation Method	Water Level (cm)	Salinity (ppt)	Flow though Haulover Canal (m³/s, m/s)
1) Historical	4.66(3.43)	3.85(3.27)	59.63(-0.02)
2) Percentile	4.66(3.44)	3.60(3.00)	60.00(0.26)
3) Spatial Percentile	4.70(3.48)	3.40(2.79)	59.49(0.37)
4) Belle Glade	4.74(3.52)	3.45(2.85)	59.45(0.10)

4.3.4 Non-Tidal Boundary Condition Importance

To determine which non-tidal boundary conditions are most important to IRL modeling, a series of 1998 simulations were performed with different boundary conditions removed (Table 4.15). These simulations show that wind is the most important factor in simulating water level and flow through Haulover Canal while discharge is the most important factor in simulating salinity transport. Discharge is moderately important in simulating flow through Haulover Canal and precipitation/evaporation are moderately important in simulating salinity transport.

4.3.5 Bottom Roughness

As shown in Equation 2.28, the bottom drag coefficient is defined as a function of the bottom roughness, z_o. The default value of bottom roughness, z_o, used for IRL simulations is a constant 0.4 cm over the entire domain. Because of the numerous different bottom types in the lagoon, a constant value may not be appropriate. Using

Table 4.15: The effect of removing selected boundary conditions on the accuracy of simulated water level, salinity and flow through Haulover Canal. Values shown are average RMS errors for all available stations. Values shown in parenthesis for water level exclude the Ponce de Leon Station (FDEP #872-1147) and Mosquito Lagoon Station (FDEP #872-1164) while values shown in parenthesis for salinity exclude the Sebastian River Station (USGS #275017080295600). Mean flow values are shown in parenthesis for flow through Haulover Canal and can be compared to the measured mean of 0.25 m^3/s with a positive value indicating flow into Mosquito Lagoon.

Boundary Condition Removed	Water Level (cm)	Salinity (ppt)	Flow though Haulover Canal (m^3/s, m/s)
None	4.70(3.48)	3.40(2.79)	59.49(0.37)
Wind	5.80(4.74)	4.66(4.13)	92.64(-5.55)
Precip./Evap.	4.80(3.61)	5.26(4.79)	60.68(-1.11)
Discharge	5.33(4.21)	7.36(6.01)	59.63(-3.24)

the sediment bottom type map (Sun, 2001) and seagrass coverage maps for the lagoon, a spatially varying bottom roughness map was developed (Figure 4.12 using the values of bottom roughness described in Table 4.16.

Table 4.16: Values of bottom roughness, z_o, used in estimating the spatial varying bottom roughness map.

	Bottom Type	z_o (cm)
Sediment	Silts or Clay	0.1
	Fine and Medium	0.3
	Coarse	0.5
	Very Coarse	0.7
Seagrass	Partial Coverage	1.0
	Patchy Coverage	1.5
	Total Coverage	2.0

To determine whether a spatially varying bottom roughness or a constant bottom roughness produces better simulated circulation and transport, a series of 1998 simulations were performed using several constant bottom roughness values and the spatially varying bottom roughness described previously (Table 4.17). The results

Bottom
Roughness
(cm)

2
1.75
1.5
1.25
1
0.75
0.5
0.25
0
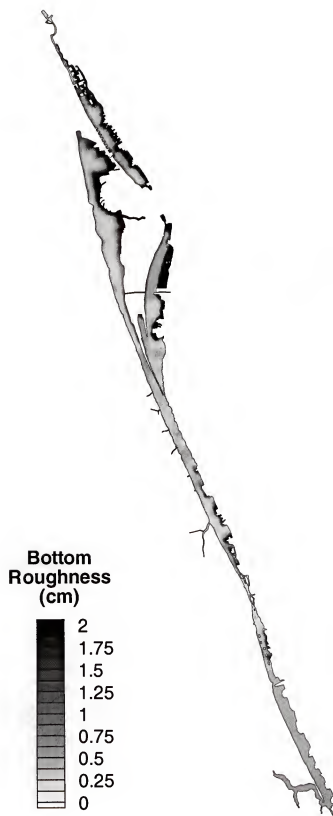
Figure 4.12: A map of estimated spatially varying bottom roughness, z_o, in the IRL.

show that simulations with the spatially varying bottom roughness have slightly smaller errors in water level and slight larger errors in salinity and flow through Haulover Canal. Since the varying bottom roughness does not improve the simulated circulation and transport significantly, the simpler constant bottom roughness of 0.4 cm is used for all simulations.

Table 4.17: The effect of varying bottom roughness,z_o, on the accuracy of simulated water level, salinity and flow through Haulover Canal. Values shown are average RMS errors for all available stations. Values shown in parenthesis for water level exclude the Ponce de Leon Station (FDEP #872-1147) and Mosquito Lagoon Station (FDEP #872-1164) while values shown in parenthesis for salinity exclude the Sebastian River Station (USGS #275017080295600). Mean flow values are shown in parenthesis for flow through Haulover Canal and can be compared to the measured mean of 0.25 m^3/s with a positive value indicating flow into Mosquito Lagoon.

Bottom Roughness (cm)	Water Level (cm)	Salinity (ppt)	Flow though Haulover Canal (m^3/s, m/s)
0.05	5.28(4.08)	3.42(2.88)	76.81(-2.23)
0.1	5.08(3.87)	3.38(2.82)	70.59(-1.65)
0.2	4.89(3.67)	3.38(2.80)	64.68(-0.85)
0.4	4.70(3.48)	3.40(2.79)	59.49(0.37)
1	4.51(3.30)	3.53(2.88)	54.69(2.52)
2	4.43(3.23)	3.67(3.01)	53.17(4.50)
Spatially Varying	4.48(3.21)	3.52(2.91)	54.78(-0.31)

4.3.6 Horizontal Diffusion Coefficient

As was shown in Equations 2.2 and 2.2, the sub-grid scale motion is estimated with a horizontal diffusion coefficient, A_H. A default value of 10,000 cm^2/s was used for IRL simulations; however, to determine the effect of varying the horizontal diffusion coefficient on the accuracy of the simulated circulation and transport, several 1998 simulations were performed using different coefficient values (Table 4.18). The simulations show little difference in water level and salinity and only a slight difference in flow through Haulover Canal; thus, the default value of 10,000 cm^2/s is appropriate for IRL simulations.

Table 4.18: The effect of varying horizontal diffusion, A_H, on the accuracy of simulated water level, salinity and flow through Haulover Canal. Values shown are average RMS errors for all available stations. Values shown in parenthesis for water level exclude the Ponce de Leon Station (FDEP #872-1147) and Mosquito Lagoon Station (FDEP #872-1164) while values shown in parenthesis for salinity exclude the Sebastian River Station (USGS #275017080295600). Mean flow values are shown in parenthesis for flow through Haulover Canal and can be compared to the measured mean of 0.25 m^3/s with a positive value indicating flow into Mosquito Lagoon.

Horizontal Diffusion Coefficient (cm^2/s)	Water Level (cm)	Salinity (ppt)	Flow though Haulover Canal (m^3/s, m/s)
5,000	4.71(3.48)	3.41(2.81)	59.90(-0.07)
10,000	4.70(3.48)	3.40(2.79)	59.49(0.37)
20,000	4.70(3.49)	3.38(2.76)	58.77(1.21)
50,000	4.72(3.52)	3.35(2.74)	56.85(3.52)

4.3.7   Vertical Grid Resolution

Except in regions close to major tributaries, the relatively shallow IRL is a well-mixed estuary. Thus, simulations are normally performed with 4 vertical layers. However, to study the effect of varying the number of vertical layers used by the CH3D model, 1998 simulations were performed with 1, 4 and 8 vertical layers (Table 4.19). When a simulation is performed using only 1 vertical layer, the water level and flow through Haulover Canal are adversely effected versus the standard 4 layer simulation presumably because of the poorer representation of bottom friction in the 2-d model. The 8 layer simulation shows a slight improvement in water level and salinity transport excluding the Sebastian River Station; however, this improvement comes at the cost of the model taking twice as long to perform the same simulation. The largest improvement in simulated salinity achieved when using 8 layers are used is at the Sebastian River Station. While the simulated surface level salinity RMS error changes little, the simulated lower level salinity RMS error improves from 11 to 6 ppt. However, because of the higher cost of performing 8 layers simulations

coupled with only a marginal improvement in simulated results, 4 layer simulations are deemed appropriate for the IRL.

Table 4.19: The effect of the number of vertical layers on the accuracy of simulated water level, salinity and flow through Haulover Canal. Values shown are average RMS errors for all available stations. Values shown in parenthesis for water level exclude the Ponce de Leon Station (FDEP #872-1147) and Mosquito Lagoon Station (FDEP #872-1164) while values shown in parenthesis for salinity exclude the Sebastian River Station (USGS #275017080295600). Mean flow values are shown in parenthesis for flow through Haulover Canal and can be compared to the measured mean of 0.25 m^3/s with a positive value indicating flow into Mosquito Lagoon.

Vertical Layers	Water Level (cm)	Salinity (ppt)	Flow though Haulover Canal (m^3/s, m/s)
1	6.42(5.25)		113.21(0.49)
4	4.70(3.48)	3.40(2.79)	59.49(0.37)
8	4.63(3.41)	2.98(2.61)	60.81(2.44)

4.3.8   Grid and Bathymetry Modification

Besides adjusting boundary conditions and model coefficients, it is also interesting to see how varying the model grid and bathymetry affects simulated circulation and transport within the lagoon. Because simulating the flow through Haulover Canal is crucial to simulating salinity in the northern IRL, it is useful to study how the flow through through Haulover Canal is affected by its cross-sectional area. As was described in Table 2.3, the cross-sectional area in Haulover Canal was fixed to a constant 201 m^2 (NAVD88). Several 1998 simulations were performed to determine how varying the cross-sectional area of Haulover Canal affects simulated water level, salinity transport and flow through the canal (Table 4.20). The results show that the simulated water level and salinity transport are not affected by a change in Haulover Canal cross-sectional area. The mean flow through the canal is also minimally affected; however, the RMS error of the simulated flow rate does improve slightly with

a more restrictive canal. Since simulated results do not improve greatly when the cross-sectional area is modified, the 201 m² is used for all subsequent simulations.

Table 4.20: The effect of changing the cross sectional area, A, of Haulover Canal (by varying its depth, H) on the accuracy of simulated water level, salinity and flow through Haulover Canal. Values shown are average RMS errors for all available stations. Values shown in parenthesis for water level exclude the Ponce de Leon Station (FDEP #872-1147) and Mosquito Lagoon Station (FDEP #872-1164) while values shown in parenthesis for salinity exclude the Sebastian River Station (USGS #275017080295600). Mean flow values are shown in parenthesis for flow through Haulover Canal and can be compared to the measured mean of 0.25 m³/s with a positive value indicating flow into Mosquito Lagoon.

Haulover Canal Depth/Area (NAVD88)	Water Level (cm)	Salinity (ppt)	Flow though Haulover Canal (m³/s, m/s)
H=400 cm / A=181 m²	4.69(3.47)	3.41(2.80)	56.52(0.42)
H=445 cm / A=201 m²	4.70(3.48)	3.40(2.79)	59.49(0.37)
H=490 cm / A=221 m²	4.71(3.50)	3.40(2.79)	64.17(0.30)

As was discussed in Chapter 2, the IRL fine grid bathymetry (Figure 2.9) was developed by interpolating measured bathymetric data onto the fine grid system followed by a simple smoothing scheme. To study how the accuracy of grid bathymetry affects simulated circulation and transport, several 1998 simulations were performed using grid systems with modified bathymetric data. The first simulation used a modified grid bathymetry which had a minimum depth of 2 m. This simulation was performed to determine how important the "shallowness" of the lagoon is to its circulation and transport. The second simulation used a bathymetry that included an artificially dredged channel through the entire lagoon. Using the Florida Inland Navigation District (FIND) Survey 96 of the Atlantic Intracoastal Waterway (ICW), an ICW was included in the find grid bathymetry at a depth of 3.75 m (NAVD88). Since the fine grid system is too coarse to adequately resolve the ICW, forcing a channel into the grid system will over-estimate the cross-sectional areas within the lagoon. Table 4.21 compares the results of the year-long simulations performed using the modified

grid bathymetry. While forcing an ICW into the fine grid system had little effect, imposing a 2 m minimum cell depth slightly worsened simulated salinity and greatly worsened both water level and flow through Haulover Canal comparisons. Since neither modified grid system improved the 1998 simulation, the standard bathymetry is used for all subsequent simulations.

Table 4.21: The effect of modifying grid bathymetry on the accuracy of simulated water level, salinity and flow through Haulover Canal. Values shown are average RMS errors for all available stations. Values shown in parenthesis for water level exclude the Ponce de Leon Station (FDEP #872-1147) and Mosquito Lagoon Station (FDEP #872-1164) while values shown in parenthesis for salinity exclude the Sebastian River Station (USGS #275017080295600). Mean flow values are shown in parenthesis for flow through Haulover Canal and can be compared to the measured mean of 0.25 m³/s with a positive value indicating flow into Mosquito Lagoon. The normal minimum cell depth is 1 to 1.5 m. The Intra-Coastal Waterway (ICW) simulation uses a different grid system with a fixed-depth ICW (3.75 m) in the cells which contain the ICW. All depths are with respect to NAVD88.

Simulation Description	Water Level (cm)	Salinity (ppt)	Flow though Haulover Canal (m³/s, m/s)
Standard Bathymetry	4.70(3.48)	3.40(2.79)	59.49(0.37)
Minimum Depth of 2 m	5.98(4.96)	3.34(2.93)	77.80(3.77)
Deeper ICW	4.71(3.49)	3.42(2.80)	60.91(-0.10)

To study how modifying natural and man-made topographic features within the lagoon will change circulation and salinity transport, two 1998 simulations were performed. The first simulation used a modified IRL fine grid system which had all representation of the natural islands in Mosquito Lagoon removed (Figure 4.13). The second simulation used a grid with all representations of the man-made causeways removed.

Table 4.22 illustrates how the accuracy of the simulation changed during these two simulations. Without the islands in Mosquito Lagoon, the water level in Mosquito

Depth (cm NAVD88)

0   50   100 150 200 250 300 350 400 450 500
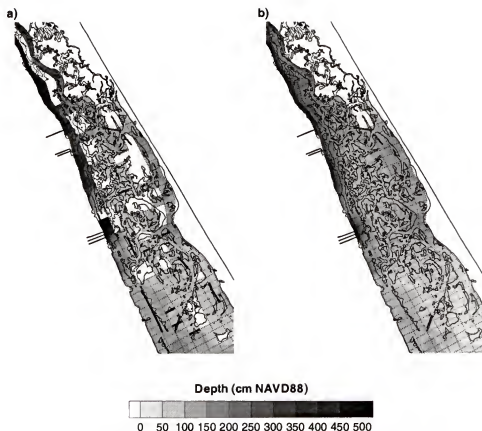
Figure 4.13: A comparison between a) the IRL fine grid and b) the IRL fine grid with the islands in Mosquito Lagoon removed.

Lagoon is lowered which draws water up from the southern IRL into Mosquito Lagoon. The mean flow from Haulover Canal into Mosquito Lagoon is increased to 14 m^3/s, which leads to increased RMS errors for simulated salinity.

Comparisons between the error analysis of the 1998 simulations of circulation and transport, with and without the man-made causeways, showed little change when the causeways were removed. Since the causeways are large impediments to flow along the length of the lagoon, this result seems surprising. However, by error analysis alone, it is not possible to conclude that the causeways do not affect salinity transport in the lagoon. Further study on this issue is needed.

Simulated residual surface salinity calculated from June 30, 1998 until July 30, 1998 shows the effect of removing the causeways more clearly (Figure 4.14). With

Table 4.22: The effect of removing the islands in Mosquito Lagoon and removing causeways resolved in the IRL fine grid system on the accuracy of simulated water level, salinity and flow through Haulover Canal. Values shown are average RMS errors for all available stations. Values shown in parenthesis for water level exclude the Ponce de Leon Station (FDEP #872-1147) and Mosquito Lagoon Station (FDEP #872-1164) while values shown in parenthesis for salinity exclude the Sebastian River Station (USGS #275017080295600). Mean flow values are shown in parenthesis for flow through Haulover Canal and can be compared to the measured mean of 0.25 m³/s with a positive value indicating flow into Mosquito Lagoon.

Simulation Description	Water Level (cm)	Salinity (ppt)	Flow though Haulover Canal (m³/s, m/s)
With Islands and Causeways	4.70(3.48)	3.40(2.79)	59.49(0.37)
Without Islands	5.44(4.09)	4.15(3.65)	72.08(14.34)
Without Causeways	4.67(3.44)	3.35(2.74)	59.95(0.64)

causeways present in the fine grid system, the low salinity water migrating up the eastern IRL shoreline is blocked by the causeway, whereas when the causeways are removed from the grid system, the low salinity water travels northward unimpeded. Thus, the causeways cause significant small scale changes in local salinity transport which the previously mentioned error analysis did not show. To more accurately determine the effect of causeway removal, more spatially and temporally dense measured salinity data are necessary since the salinity data used for model comparison are spatially sparse and most of the FDEP salinity stations used for model comparisons are located on the actual causeways being studied, possibly biasing the results.

4.4 The 1998 Simulation

For the 1998 simulation, the original fine grid and bathymetry, with all causeways represented, are used along with the boundary conditions and model parameters described in Table 4.23. Water level, salinity and flow rate are compared both qualitatively and quantitatively with measured data, where available.
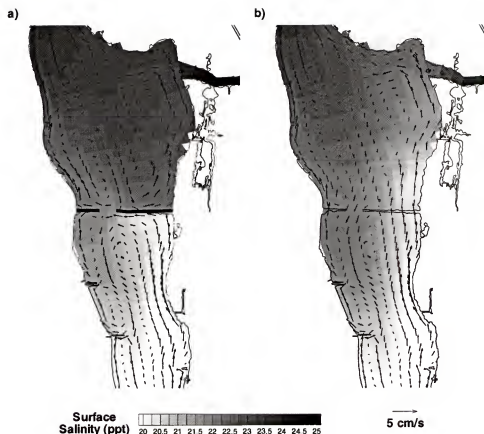
Figure 4.14: A comparison between the simulated residual surface salinity and circulation, in the vicinity of the NASA Causeway between June 30, 1998 and July 30, 1998, using the IRL fine grid system a) with the NASA causeway represented (thick dark lines) and b) without the NASA causeway represented.

4.4.1 Water Level

Comparisons between simulated and measured mean water level for the 1998 simulation are shown in Table 4.24.

Calculated RMS errors between simulated and measured water level during the 1998 simulation are shown in Table 4.25. Year-long and daily comparisons between simulated and measured water level for all available data appear in Appendix J. Spectral density comparisons between simulated and measured water level appear in Appendix K.

Table 4.23: A summary of boundary conditions and model parameters used in the 1998 simulation.

Boundary Condition or Model Parameter	Value
Ponce Inlet Tidal Forcing	Measured - 290 s, MWL - 10.5 cm
Sebastian Inlet Tidal Forcing	Measured
Ft. Pierce Inlet Tidal Forcing	Measured - 120 s
St. Lucie Tidal Forcing	Regenerated
Wind Speed and Direction	Measured
Precipitation	Measured
Evaporation	Estimated (Spatial Percentile)
Fresh Water Discharge	Simulated and Measured (24)
Bottom Roughness (z_o)	Constant (0.4 cm)
Horizontal Diffusion Coefficient (A_H)	Constant (10000 cm^2/s)
Vertical Layers	4

Residuals of simulated water level calculated every 30 days for the first and last six months of 1998 can be seen in Figures 4.15a and 4.15b, respectively.

To determine if the model correctly simulates the various tidal frequencies within the lagoon, a harmonic analysis is performed on both measured and simulated water level. The software chosen to perform the harmonic analysis is the Institute of Ocean Sciences Tidal Package (IOS TP) (Foreman, 1996). Using the methodologies described in Godin (1972) and Godin and Taylor (1973), the IOS TP analyzes hourly data for a given length of time and produces amplitudes and phases for 69 constituents and up to 77 additional shallow water constituents. The IOS TP also has the ability to handle gaps in the data record and can regenerate the input data record based on the determined constituents.

Originally, the IOS TP input record and regenerated output only allowed four characters (including any negative signs) for each data record. Due to the yearly range of the input data, the accuracy of these files is limited to 0.5 cm for water level. In the northern IRL and Banana River, where tidal amplitudes are on the order of centimeters, four characters do not provide sufficient accuracy of the harmonic

Table 4.24: A comparison between simulated and measured mean water levels. The measured mean water level is calculated using observations collected at six minute intervals while the simulated mean water levels are calculated using hourly intervals.

Station		Measured	Simulated	Relative	Range	% Error
Name	Number	(cm)	(cm)	Error (cm)	(cm)	
Ponce de Leon Inlet	872-1147 (FDEP)	-24.32	-35.00	-10.68	196.10	5.45
Mosquito Lagoon	872-1164 (FDEP)	-22.94	-35.94	-13.00	157.40	8.26
Hanlover Canal	02248380 (USGS)	-23.59	-27.75	-4.16	57.40	7.25
Titusville	872-1456 (FDEP)	-22.18	-27.69	-5.52	59.50	9.27
Merritt Causeway (East)	872-1647 (FDEP)	-25.58	-27.83	-2.25	42.60	5.27
Merritt Causeway (West)	872-1648 (FDEP)	-24.06	-28.12	-4.06	50.40	8.05
Banana River	872-1789 (FDEP)	-24.46	-28.26	-3.80	64.00	5.94
Melbourne Causeway	872-1843 (FDEP)	-24.09	-27.93	-3.84	64.40	5.96
Sebastian Inlet	872-2004 (FDEP)	-36.62	-36.62	0.00	149.00	0.00
Wabasso	02251800 (USGS)	-31.40	-30.58	0.82	106.80	0.77
Vero Bridge	872-2125 (FDEP)	-28.93	-29.57	-0.64	95.80	0.67
Ft. Pierce Causeway	872-2208 (FDEP)	-31.20	-35.19	-3.99	142.70	2.80
Ft. Pierce Inlet	872-2213 (FDEP)	-36.20	-36.10	0.10	160.30	0.06
St. Lucie Inlet	872-2375 (FDEP)	-36.21	-36.21	0.00	137.10	0.00
Average		-27.98	-31.63	-3.65	105.96	3.44
Average (w/o 872-1147, 872-1164)		-28.71	-30.99	-2.28	94.17	2.42
Average (w/o Inlet Stations)		-25.84	-29.89	-4.04	84.10	4.81

Water Level (cm NAVD88)
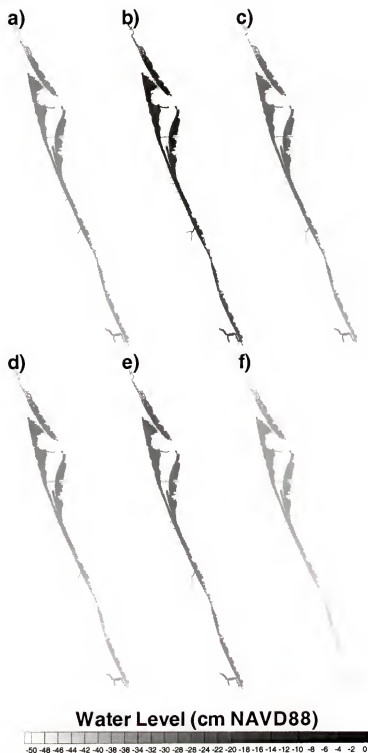
Figure 4.15a: Residual simulated water level calculated every 30 days during the first 6 months of 1998. Residual values are shown ending on the following days: a) January 31, b) March 3, c) April 1, d) May 1, e) May 31 and f) June 30.

Water Level (cm NAVD88)

-50 -48 -46 -44 -42 -40 -38 -36 -34 -32 -30 -28 -26 -24 -22 -20 -18 -16 -14 -12 -10 -8 -6 -4 -2 0
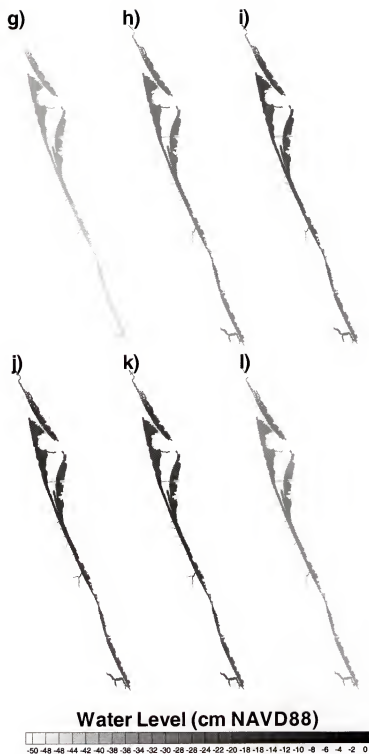
Figure 4.15b: Residual simulated water level calculated every 30 days during the last 6 months of 1998. Residual values are shown ending on the following days: g) July 30, h) August 29 i) September 28, j) October 28, k) November 27 and l) December 27.

Table 4.25: Calculated RMS errors between simulated and measured water level.

Station		RMS Error	Range	% Error
Name	Number	(cm)	(cm)	
Ponce de Leon Inlet	872-1147 (FDEP)	10.72	196.10	5.47
Mosquito Lagoon	872-1164 (FDEP)	13.33	157.40	8.47
Haulover Canal	02248380 (USGS)	5.55	57.40	9.67
Titusville	872-1456 (FDEP)	5.99	59.50	10.07
Merritt Causeway (East)	872-1647 (FDEP)	2.80	42.60	6.57
Merritt Causeway (West)	872-1648 (FDEP)	4.81	50.40	9.54
Banana River	872-1789 (FDEP)	4.21	64.00	6.58
Melbourne Causeway	872-1843 (FDEP)	4.22	64.40	6.56
Sebastian Inlet	872-2004 (FDEP)	0.00	149.00	0.00
Wabasso	02251800 (USGS)	3.05	106.80	2.86
Vero Bridge	872-2125 (FDEP)	2.78	95.80	2.90
Ft. Pierce Causeway	872-2208 (FDEP)	6.44	142.70	4.51
Ft. Pierce Inlet	872-2213 (FDEP)	1.94	160.30	1.21
St. Lucie Inlet	872-2375 (FDEP)	0.00	137.10	0.00
Average		4.70	105.96	5.31
Average (w/o 1147, 1164)		3.48	94.17	5.04
Average (w/o Inlets)		5.32	84.10	6.77

analysis and regeneration. This can be easily seen when trying to regenerate the input data record. Thus, the IOS TP was modified to allow six characters (including any negative signs) for each data record improving the accuracy of the input record and regenerated output to 0.005 cm. The calculated amplitudes and phases changed on the order of several percent for most constituents, while the appearance of the regenerated data record was greatly improved.

Comparisons show that the model is able to capture both the amplitudes (Figure 4.17) and phases (Figure 4.17) well.
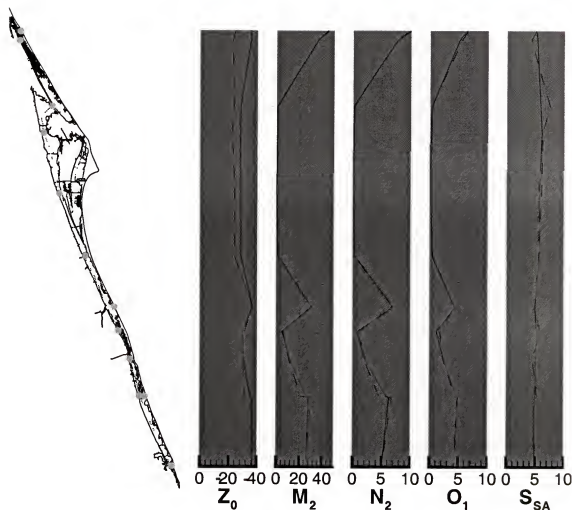
Figure 4.16: A comparison between simulated and measured amplitude for four harmonic constituents and the mean water level (Z_o). Solid lines correspond to simulated amplitudes while dashed lines correspond to measured amplitudes. Amplitudes are given in centimeters. From North to South, the stations used to compare are FDEP #872-1147, FDEP #872-1164, USGS #02248380, FDEP #872-1456, FDEP #872-1648, FDEP #872-1843, FDEP #872-2004, USGS #02251800, FDEP #872-2125, FDEP #872-2208, FDEP #872-2213, and FDEP #872-2375.
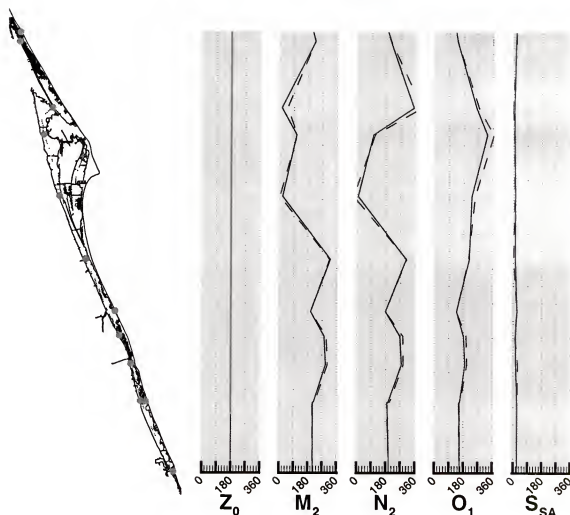
Figure 4.17: A comparison between simulated and measured phase for four harmonic constituents and the mean water level (Z_o). Solid lines correspond to simulated phases while dashed lines correspond to measured phases. Phases are relative to Greenwich and are given in degrees. From North to South, the stations used to compare are FDEP #872-1147, FDEP #872-1164, USGS #02248380, FDEP #872-1456, FDEP #872-1648, FDEP #872-1843, FDEP #872-2004, USGS #02251800, FDEP #872-2125, FDEP #872-2208, FDEP #872-2213, and FDEP #872-2375.

Although no measured velocity data were available for model comparison during the 1998 study, a portion of the northern IRL (Figure 4.18) was studied to determine the relationship between bathymetric features and wind circulation patterns. The northern IRL was chosen because the semi-diurnal and diurnal constituent amplitudes are only of marginal practical importance with 92-99% of the total variance being composed of non-tidal, low frequency signals (Smith, 1987).

Simulated vertically averaged velocities in the study area during the first and last six 30-day periods of 1998 are shown in Figures 4.19a and 4.19b, respectively As expected, the circulation patterns in the non-tidal environment vary with wind magnitude and direction. These figures also illustrate the effect that the causeways have on residual circulation patterns. With the causeways restricting flow to pass through a small area, artificial gyres are formed trapping any pollutant or floating body that might be present.

The time periods can be divided into four groups based on average wind speed and direction and circulation characteristics: strong southerly wind (b, d, e, f, g, and i), weak southerly wind (a and c), easterly wind (h, j, and k) and northernly wind (l). With a strong southerly wind, the strong northward residual flow in shallow waters (2 to 3 cm/s) near the shore is balanced by a strong southerly residual flow in the channel (2 to 3 cm/s). A weak southerly wind leads to a very weak northerly nearshore flows (0 to 0.5 cm/s) and small flow in varying directions in the channel (-0.5 to 0.5 cm/s). Easterly winds do not lead to any well defined circulation patterns. Northern winds lead to a strong southward residual flow in the shallow waters (2 to 3 cm/s) near the shore; however, the flow in the channel remains small (< 0.5 cm/s). The southward flow is balanced by flow into the northern IRL through Haulover Canal which can be seen in the upper right hand corner of the figure.

Figure 4.18: Location of the residual velocity study area.
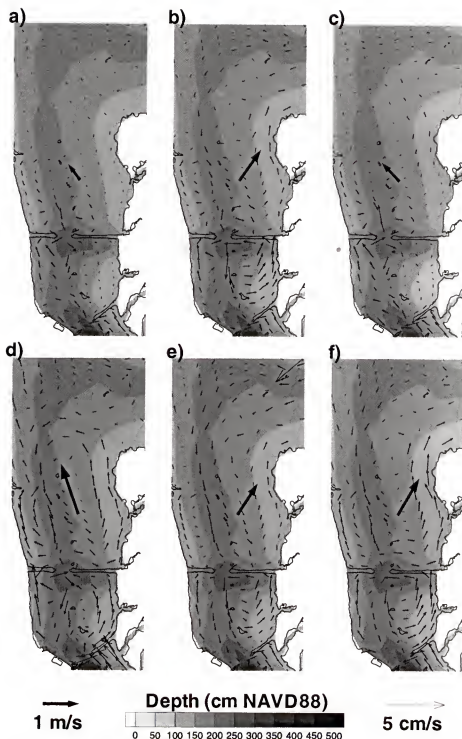
Figure 4.19a: Residual vertically averaged simulated velocities every 30 days during the first 6 months of 1998. Current velocities are shown with thin arrows while the thick arrow is the average wind speed and direction during the same 30 days. The Train Causeway is slightly south of center and Brewer Causeway is near the very bottom. Residual values are shown ending on the following days: a) January 31, b) March 3, c) April 1, d) May 1, e) May 31 and f) June 30.

Depth (cm NAVD88)

1 m/s

0 50 100 150 200 250 300 350 400 450 500

5 cm/s
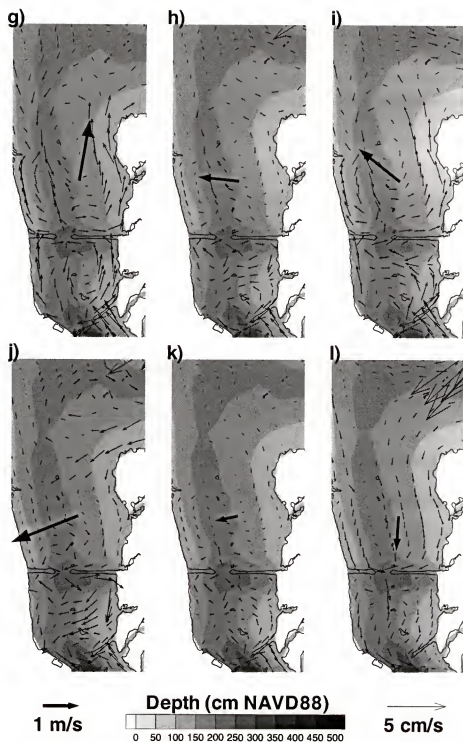
Figure 4.19b: Residual vertically averaged simulated velocities every 30 days during the first 6 months of 1998. Current velocities are shown with thin arrows while the thick arrow is the average wind speed and direction during the same 30 days. The Train Causeway is slightly south of center and Brewer Causeway is near the very bottom. Residual values are shown ending on the following days: g) July 30, h) August 29 i) September 28, j) October 28, k) November 27 and l) December 27.

4.4.3   Flow Rate

The only long-term data set available within the lagoon measuring flow rate was located in Haulover Canal. The 1998 maximum measured flow was 366.71 m^3/s and the minimum was -310.08 m^3/s, which results in a range of 676.79 m^3/s. The measured mean flow rate during 1998 simulation is 0.25 m^3/s while the simulated flow rate is 0.37 m^3/s (0.02%) where a positive value indicates flow into Mosquito Lagoon. The RMS error between simulated and measured flow rate is 59.49 m^3/s (8.79%). During 1998, approximately 10% of the measured flow rate data in Haulover Canal were missing. Thus, comparisons between simulated and measured flow rate are only performed at times when measured data are present.

Comparisons between simulated and measured flow rates in Haulover Canal during four days of the 1998 simulation are shown in Figure 4.20 while comparisons between simulated and measured flow rates during the four USGS sampling events at the four inlets appear in Appendix L.

4.4.4   Salinity

Before baroclinic simulations are performed it is necessary to carefully generate an initial salinity field because, in general, salinity is much slower to respond to initial transients than water level or currents. The initial salinity field for the 1998 simulation is generated by "spinning up" a specified salinity field for a certain length of time and then using the values at the end of the simulation as the initial values for the 1998 simulations. This process is discussed below.

First, a salinity field is created by linearly interpolating measured salinity into the IRL grid. The first salinity measurements taken during 1998 by the FDEP and IRL-WQMN are used to create this salinity field (Table 4.26). Next, a salinity assimilation term is added to the salinity transport equation in the model. This term forces the salinity in selected cells to approach a specific value, in this case, the initial value. The eight FDEP measured salinity values shown in Table 4.26 are
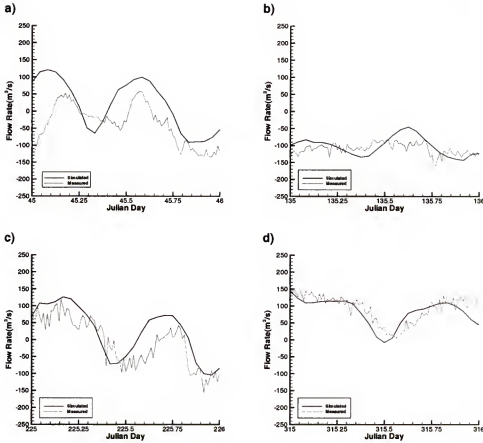
Figure 4.20: Comparisons between simulated and measured flow rate past the Haulover Canal Station (USGS #02248380) during four days of the 1998 simulation: a) February 14 (45), b) May 15 (135), c) August 13 (225) and d) November 11 (315). The simulated flow rate is shown with a solid line while the measured flow rate is shown a dotted line. A positive flow indicates flow from the northern IRL into Mosquito Lagoon.

chosen to be assimilated into the simulation. These values were chosen because they were measured closest to the January 1 starting date of the 1998 simulations. The assimilation term takes the following form

$$-\frac{S_{i,j,k}^n - (S_A)_{i,j,k}}{T} \tag{4.2}$$

where $S_{i,j,k}^n$ is the simulated salinity at the nth time level, $(S_A)_{i,j,k}$ is the value to be assimilated, in this case, the initial salinity value, $(S_i)_{i,j,k}$, and T is the assimilation period, 30 days. It can be seen, that for $S_{i,j,k}$ not equal to $(S_i)_{i,j,k}$ a force is created

driving $S_{i,j,k}$ toward the value of $(S_i)_{i,j,k}$. Because of the addition of the salinity assimilation term, the salinity at the eight FDEP sites at the end of a simulation will be close to their values at the beginning of the simulation.

Next, a 60 day simulation is performed using the linearly interpolated salinity field with the salinity assimilation term applied to the cells which contain the eight FDEP measured salinity values. This simulation begins November 2, 1997 and continues for 60 days until January 1, 1998. During the simulation, measured data (as discussed previously) is used for the tidal and wind forcing. Because neither measured precipitation, evaporation nor discharge data are available before 1998, these data are estimated using the November 2, 1998 to January 1, 1999 measured values. This period in 1998 is chosen so that the seasonal aspects of discharge, precipitation and evaporation will be imposed.

Finally, the hydrodynamic and salinity fields at the end of the 60-day simulation are saved and used as an initial hydrodynamic and salinity field for the 1998 simulations. The initially interpolated water level field and the water level field at the end of the 60-day simulation are shown in Figure 4.21 while the initial interpolated surface salinity field and the surface salinity field at the end of the 60-day simulation are shown in Figure 4.22.

The salinity is measured at two locations in the water column referred to as "upper" and "lower". The vertical positions of the salinity measurements taken by the FDEP are given in Table 4.27. For this study, four vertical layers are used by the CH3D model. An average of the simulated salinity at levels two and three (50% of the total depth) are compared with the salinity measured by the upper sensor and the first level (12.5% of total depth) is compared to the lower sensor.

Calculated RMS errors between the simulated and measured salinity during the 1998 simulation have been divided into yearly (Table 4.28), monthly (Table 4.29)

Table 4.26: The average vertical salinity for the first collection time in 1998.

Station ID	Agency	Salinity (ppt)	Collection Date
872-1164	FDEP	15.0	January 18
V05	IRL-WQMN	30.2	January 13
V11	IRL-WQMN	27.7	January 13
V17	IRL-WQMN	27.5	January 13
TBC	IRL-WQMN	0.2	January 12
I02	IRL-WQMN	26.0	January 14
ML02	IRL-WQMN	22.5	January 13
872-1456	FDEP	24.8	January 1
I07	IRL-WQMN	24.0	January 14
I10	IRL-WQMN	22.0	January 13
B02	IRL-WQMN	13.0	January 14
I13	IRL-WQMN	20.1	January 13
B04	IRL-WQMN	14.9	January 14
872-1647	FDEP	14.6	January 1
872-1648	FDEP	14.4	January 1
B06	IRL-WQMN	13.7	January 14
I16	IRL-WQMN	19.2	January 13
B09	IRL-WQMN	12.0	January 14
I18	IRL-WQMN	16.2	January 13
HUS	IRL-WQMN	6.0	January 12
872-1789	FDEP	12.8	January 31
I21	IRL-WQMN	15.2	January 13
EGU	IRL-WQMN	5.1	January 12
872-1843	FDEP	15.8	January 1
CCU	IRL-WQMN	2.5	January 12
I23	IRL-WQMN	12.1	January 13
GUS	IRL-WQMN	0.5	January 12
I27	IRL-WQMN	13.5	January 13
SUS	IRL-WQMN	3.0	January 12
IRJ01	IRL-WQMN	25.0	January 14
IRJ10	IRL-WQMN	22.1	January 13
IRJ04	IRL-WQMN	23.7	January 14
IRJ05	IRL-WQMN	23.3	January 13
VMC	IRL-WQMN	0.5	January 12
872-2125	FDEP	22.0	January 1
IRJ07	IRL-WQMN	22.7	January 13
IRJ12	IRL-WQMN	23.0	January 14
VSC	IRL-WQMN	0.7	January 12
872-2208	FDEP	28.1	January 1

a) **b)**

Water Level (cm NAVD88)

-50 -48 -46 -44 -42 -40 -38 -36 -34 -32 -30 -28 -26 -24 -22 -20 -18 -16 -14 -12 -10 -8 -6 -4 -2 0

Figure 4.21: Initial water level fields a) interpolated using the measured water level data and b) after a 60 day "spin up" simulation.

Surface Salinity (ppt)

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35

Figure 4.22: Initial surface salinity fields a) interpolated using the measured salinity data and b) after a 60 day "spin up" simulation which fixed the salinity values at several locations.

Table 4.27: Vertical positions of FDEP salinity sensors relative to NAVD88. Water level is measured positive up while the positions of the sensors and depth are measured positive down.

Station Number	Mean Water Level (m NAVD88)	Location of Top Sensor (m NAVD88)	Location of Bottom Sensor (m NAVD88)	Measured Depth (m NAVD88)
872-1164	-0.23	5.36 (51%)	9.32 (91%)	10.27
872-1456	-0.22	3.60 (48%)	6.89 (94%)	7.32
872-1647	-0.26	3.47 (49%)	6.58 (96%)	6.83
872-1648	-0.24	5.67 (64%)	7.68 (87%)	8.75
872-1789	-0.24	4.88 (75%)	6.34 (98%)	6.46
872-1843	-0.24	3.96 (57%)	6.49 (96%)	6.74
872-2125	-0.29	3.93 (56%)	6.49 (96%)	6.77
872-2208	-0.31	3.54 (42%)	6.89 (85%)	8.02
Average	-0.25	4.30 (55%)	7.09 (92%)	7.65

and seasonal (Table 4.30) values. Year-long and daily time series comparisons between the simulated and measured salinity appear in Appendix M. Figure 4.23 shows the seasonal RMS error for salinity along the length of the lagoon.

Residuals of the surface salinity field calculated every 30 days for the first and last six months of 1998 can be seen in Figures 4.24a and 4.24b, respectively.

The following is a summary of the comparisons between simulated and measured data,

- the relative error for mean water level was 3.44%,

- the RMS error for water level was 5.31%,

- the relative error for mean flow rate through Haulover Canal was 0.02%,

- the RMS error for flow rate through Haulover Canal was 8.79%, and

- the RMS error for salinity was 18.07%.

Table 4.28: RMS errors for salinity. The vertical positions of measurement, surface and bottom, are indicated by "s" and "b", respectively.

Station		Layer	RMS Error	Range	% Error
Name	Number		(ppt)	(ppt)	
Mosquito Lagoon	872-11641 (FDEP)	s	2.98	11.4	26.10
		b	2.87	12.1	23.71
Haulover Canal	02248380 (USGS)	s	0.99	15.0	6.57
		b	1.22	17.8	6.86
Titusville	872-14561 (FDEP)	s	3.68	17.7	20.78
		b	2.10	12.9	16.30
Merritt Causeway (E)	872-16471 (FDEP)	s	1.00	8.1	12.25
		b	1.47	7.4	19.86
Merritt Causeway (W)	872-16481 (FDEP)	s	3.06	16.7	18.31
		b	1.80	11.9	15.09
Banana River	872-17891 (FDEP)	s	1.84	16.4	11.22
		b	2.40	14.7	16.39
Melbourne Causeway	872-18431 (FDEP)	s	2.97	19.3	15.43
		b	4.88	20.0	24.33
Sebastian River	275017080295600	s	6.63	34.2	19.39
	(USGS)	b	11.22	35.5	31.61
Vero Bridge	872-21251 (FDEP)	s	3.77	16.6	22.64
		b	3.57	21.6	16.54
Ft. Pierce Causeway	872-22081 (FDEP)	s	6.63	32.0	20.75
		b	3.02	17.4	17.38
Average		s	3.35	18.7	17.34
Average		b	3.46	17.1	18.81
Average			3.40	17.9	18.07
Average (w/o Seb. R.)		s	2.99	17.0	17.12
Average (w/o Seb. R.)		b	2.59	15.1	17.38
Average (w/o Seb. R.)			2.79	16.1	17.25

Table 4.29: Monthly RMS errors for salinity. The vertical positions of measurement, surface and bottom, are indicated by "s" and "b", respectively.

Station Number	Name	Layer	RMS Error (ppt)											
			Jan.	Feb.	Mar.	Apr.	May	Jun.	Jul.	Aug.	Sep.	Oct.	Nov.	Dec.
872-11641 (FDEP)	Mosquito Lagoon	s				4.48		2.65	2.20				3.59	3.09
		b				3.02		1.17	0.77			4.72	2.17	1.11
02248380 (USGS)	Haulover Canal	s				0.80	0.94	0.76	1.20	1.07	0.97	0.98	0.75	1.17
		b				0.92	0.89	0.75	2.28	0.94	0.98	1.23	0.85	1.34
872-14561 (FDEP)	Titusville	s	3.82	4.01	4.73	4.02	2.03	4.14	3.33	3.54	3.05	1.88	1.47	1.78
		b	1.81	1.66	2.81	2.15	1.20	1.91	1.99	2.22		0.71	2.35	2.40
872-16471 (FDEP)	Merritt Cswy (E)	s	0.63	0.91	1.66	0.44	0.31	0.52	0.62	1.63	1.99	1.92	0.56	1.73
		b	0.48	0.64	0.88	0.24	0.50	0.35	0.89	1.94		1.74	1.84	2.24
872-16481 (FDEP)	Merritt Cswy (W)	s	9.62			3.46	1.06	1.11		0.83	1.81	0.82	2.06	1.02
		b	4.60			0.74	4.73	2.84		1.19	2.02	1.30	0.59	0.85
872-17891 (FDEP)	Banana River	s	1.34				4.96	4.14	2.35	2.09	1.40	1.85	0.76	1.62
		b	1.89				5.20	1.29	3.56	2.84	2.18	1.39	1.27	1.06
872-18431 (FDEP)	Melbourne Causeway	s	2.18	2.65	1.91	4.94	7.78	3.96	4.44	3.18	3.37	2.17		
		b	5.27	3.57	2.77	7.41	10.41	11.58		3.86		3.34		
2750170802956000 (USGS)	Sebastian River	s	3.79	3.65	2.36	8.88	13.52	11.02	9.95	10.01	12.21	9.69	4.29	4.55
		b	11.14	10.59	5.19	12.66		7.03	11.02	14.26		2.10	11.19	9.22
872-21251 (FDEP)	Vero Bridge	s	6.59	4.42	2.90	2.17	2.91	2.24	4.14	3.80	0.90	4.30	1.92	0.98
		b	4.93	3.07	2.02			3.02	3.12	4.26	3.59	3.44	3.44	1.21
872-22081 (FDEP)	Ft. Pierce Cswy	s	6.45	11.51	11.74	7.28	3.20		1.90	2.56	8.51	2.07		
		b	2.69	3.88	3.96	3.96			0.58	1.90				
	Average		4.15	4.25	3.58	3.97	3.98	3.63	2.70	3.45	3.13	2.54	2.14	2.21

Table 4.30: Seasonal RMS error for salinity. The vertical positions of measurement, surface and bottom, are indicated by "s" and "b", respectively.

Station Name	Number	Layer	RMS Error			
			Spring	Summer	Fall	Winter
Mosquito Lagoon	872-11641 (FDEP)	s	3.21(28.12)	2.28(19.98)	3.26(28.55)	
		b	1.54(12.74)	0.78(6.42)	3.63(29.95)	
Haulover Canal	02248380 (USGS)	s	0.83(5.52)	1.05(7.02)	1.00(6.64)	1.46(9.76)
		b	0.86(4.84)	1.46(8.21)	1.16(6.52)	1.76(9.91)
Titusville	872-14561 (FDEP)	s	3.74(21.14)	3.04(17.16)	1.82(10.30)	4.05(22.89)
		b	1.76(13.68)	2.25(17.47)	2.16(16.76)	2.15(16.70)
Merritt Causeway (E)	872-16471 (FDEP)	s	0.48(5.89)	1.16(14.31)	0.83(10.26)	1.30(16.01)
		b	0.51(6.86)	1.64(22.09)	2.11(28.37)	0.84(11.35)
Merritt Causeway (W)	872-16481 (FDEP)	s	3.46(20.75)	1.45(8.69)	1.75(10.46)	7.35(44.06)
		b	0.96(8.05)	1.49(12.51)	0.81(6.81)	4.15(34.74)
Banana River	872-17891 (FDEP)	s	4.73(28.84)	2.18(13.27)	1.04(6.36)	1.79(10.89)
		b	4.96(33.88)	3.14(21.41)	1.53(10.46)	1.67(11.38)
Melbourne Causeway	872-18431 (FDEP)	s	4.08(21.18)	2.32(12.05)	1.39(7.22)	2.29(11.90)
		b	6.32(31.51)	4.09(20.41)	2.17(10.80)	4.15(20.69)
Sebastian River	275017080295600 (USGS)	s	9.70(28.39)	6.96(20.37)	3.65(10.69)	3.71(10.85)
		b	12.38(34.86)	11.98(33.73)	10.33(29.10)	9.82(27.65)
Vero Bridge	872-21251 (FDEP)	s	8.52(51.18)	3.94(23.68)	1.87(11.21)	5.37(32.26)
		b	2.79(12.93)	3.81(17.64)	3.44(15.92)	3.69(17.07)
Ft. Pierce Causeway	872-22081 (FDEP)	s	7.41(23.18)	6.47(20.23)	4.96(15.51)	7.85(24.55)
		b	3.40(19.60)	1.76(10.16)	2.10(12.11)	3.75(21.62)
Average			4.08(20.66)	3.16(16.34)	2.55(14.20)	3.73(19.68)
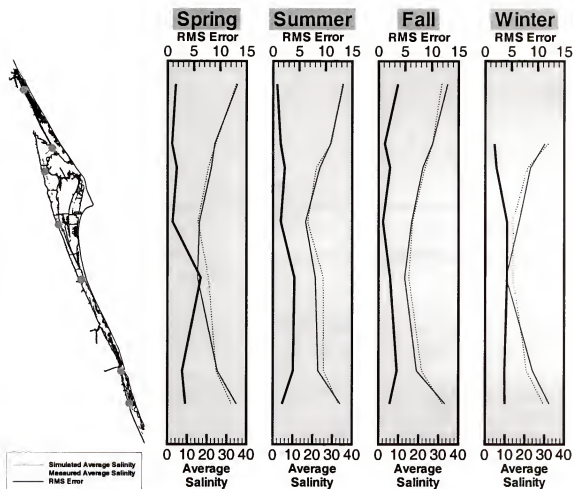
Figure 4.23: The simulated and measured seasonal salinity and the RMS error for the seasonal salinity. From North to South, the stations used to compare are FDEP #872-1164, USGS #02248380, FDEP #872-1456, FDEP #872-1648, FDEP #872-1843, FDEP #872-2125 and FDEP #872-2208

Surface Salinity (ppt)

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35
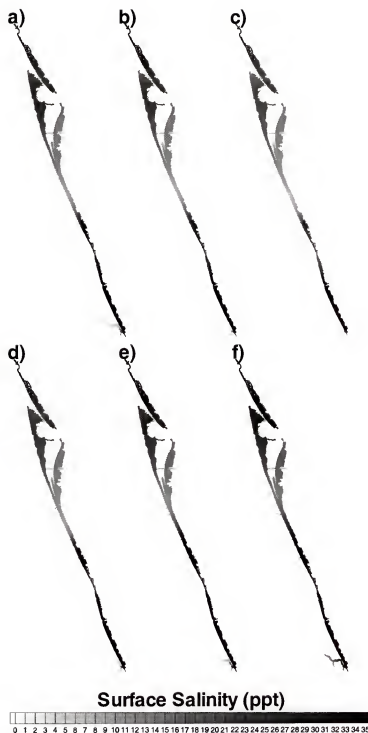
Figure 4.24a: Residual simulated surface salinity calculated every 30 days during the first 6 months of 1998. Residual values are shown ending on the following days: a) January 31, b) March 3, c) April 1, d) May 1, e) May 31 and f) June 30.

Surface Salinity (ppt)

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35
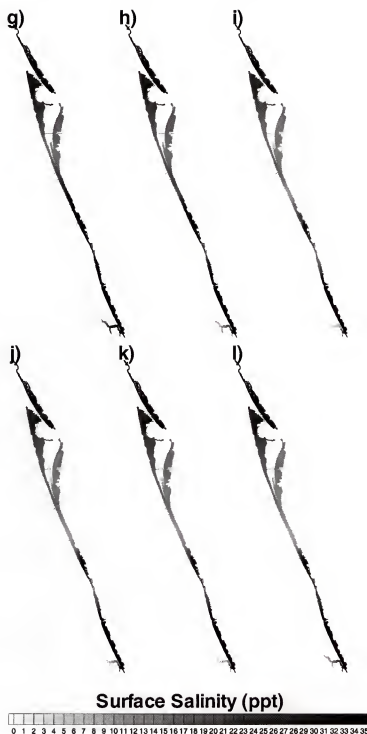
Figure 4.24b: Residual simulated surface salinity calculated every 30 days during the last 6 months of 1998. Residual values are shown ending on the following days: g) July 30, h) August 29 i) September 28, j) October 28, k) November 27 and l) December 27.

<u>4.4.5 Budgets and Conservation</u>

During the 1998 simulation, water and salt budgets were created for the IRL using fluxes of water and salt and the change in interior volume. Yearly and seasonal budgets were created for both water volume and salt mass. Positive fluxes are directed into the IRL.

The simulated water budget for the IRL using the fine grid is shown in Table 4.31. The budget shows a net evaporation (135.1×10^6 m^3) and overall a small gain in total water volume (87.6×10^6 m^3) during the year, a large amount of fresh water enters the lagoon, more than twice the total interior volume. However, a majority of this water enters through the southern IRL (Figure 4.7), specifically St. Lucie River and exits through St. Lucie Inlet. The net inflow being equal to the interior change in water volume indicates the conservative property of the simulation model.

Table 4.31: The simulated water budget for IRL using the fine grid. Values shown are in units of 10^6 m^3. The total interior water volume at the beginning of the simulation is 1531.4×10^6 m^3.

Time Period	Inlet	Discharge & Runoff	Net Precipitation & Evaporation	Net Inflow	Interior Change
Spring	-601.5	920.2	-328.0	-9.2	-9.2
Summer	-506.2	582.3	114.2	190.3	190.3
Fall	-715.9	651.4	-44.6	-109.1	-109.2
Winter	-1665.5	1557.9	123.3	15.6	15.6
Year	-3489.2	3711.8	-135.1	87.5	87.6

The simulated salt budget for the IRL using the fine grid is shown in Table 4.32. The budget shows 9.6×10^6 kg of salt exited through the inlets in the winter because of increased discharge during the same period. During the spring, 10.1×10^6 kg of salt entered because of decreased fresh water discharge coupled with a decrease in tidal outflow of water caused by an increase in evaporation. As with the

water budget, the net inflow being equal to the interior change in salt mass indicates the conservative property of the simulation model.

Table 4.32: The simulated salt budget for IRL using the fine grid. Values shown are in units of 10^6 kg. The total interior salt mass at the beginning of the simulation is 33.2×10^6 kg.

Time Period	Inlet	Discharge & Runoff	Net Precipitation & Evaporation	Net Inflow	Interior Change
Spring	10.1	0.6	0.0	10.7	10.7
Summer	-7.0	2.9	0.0	-4.1	-4.1
Fall	-1.1	3.1	0.0	2.0	2.0
Winter	-9.6	5.7	0.0	-3.9	-3.9
Year	-7.6	12.3	0.0	4.7	4.7

While the water and salt budgets show the conservative properties of the hydrodynamic and salinity models in seasonal and yearly scales, it is also useful to check the conservative properties of the model during each time iteration to make sure conservation is not violated on small time scales. Figure 4.25 shows the conservation error during the 1998 simulation. Less than 0.01% of water and salt are lost; thus demonstrating the conservative properties of the hydrodynamic and salinity models.

4.5   Parallelization Analysis

As was discussed in Chapter 2, the CH3D model was converted from a serial code to a combined serial and parallel code. A profiling analysis was performed and the procedures were ranked from most CPU intensive to the least. The subroutines which take the most CPU time were then converted into the combined serial and parallel code. Depending on the state of various cpp flags during compilation, the CPU intensive procedures can then be made to run in parallel.
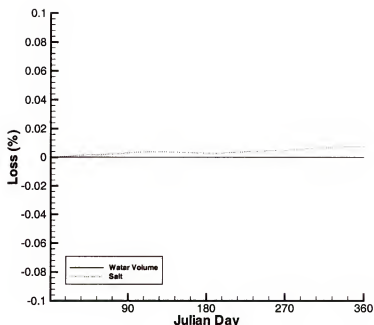
Figure 4.25: Conservation error in the boundary-fitted fine grid during the 1998 simulation.

The success of the parallelization can be measured by the accuracy of the results, speedup and efficiency. Simulations using the main CH3D models (hydrodynamics and salinity, flushing, sediment and water quality) were analyzed on three computing platforms and are discussed in this section.

4.5.1 Computing Platforms

Three computing platforms were used to test the shared memory version of CH3D (Table 4.33). These platforms represent a range of small to medium priced computing systems. The Origin 2000 is costliest ($\approx \$85,000$ in 1997), followed by the Octane $\approx \$12,000$ in 1998), and the PowerEdge($\approx \$2,000$ in 1999). While these computers are not the latest in their respective product lines, their pricepoints have not changed[10]. It is noted that prices quoted here include educational discounts.

---

[10]At the time of writing this document, the latest similarly priced models are the Silicon Graphics Origin 3400, the Silicon Graphics Octane2 and the Dell PowerEdge 1400

Table 4.33: Computing platforms tested using the shared memory version of CH3D.

Host Name	Number of Processors	Main Memory Size (MB)	Description
"ocean"	4	512	Silicon Graphics Origin 2000 195 MHz MIPS R10000 (IP27)
"seiche"	2	256	Silicon Graphics Octane 225 MHz MIPS R10000 (IP30)
"nereus"	2	256	Dell PowerEdge 1300 450 MHz Intel Pentium III

Since the majority of CPU time in a numerical model is spent performing floating point operations, the Linpack Benchmark (Dongarra, 2001) is used to compare the floating point speed of these three platforms. The single precision Linpack Benchmark for matrices of order 1000 produced speeds of 97 $Mflop/s$[11], 52 $Mflop/s$, and 61 $Mflop/s$, respectively. While the Origin 2000 performs best in this benchmark, it is noted that "results reflect only one problem area: solving dense systems of equations" (Dongarra, 2001).

4.5.2 Description of Simulations

Four sets of scenarios are used to test the parallel CH3D model. The parallel CH3D model used in these scenario tests has not been calibrated, since the primary objective of these tests is to assess the model's efficiency. The first three scenarios are #1) hydrodynamics and salinity, #2) hydrodynamics, salinity and flushing and #3) hydrodynamics, salinity, sediment and water quality. These scenarios consist of 7200 time iterations performed on the IRL fine grid. A time step of 60 s is used for Scenario #1 and #2 (a 5 day simulation); however, because of stability issues associated with the water quality model, a 30 s time step is required for Scenario #3 (a 2.5 day simulation). Scenario #4 consists of 7200 time iterations of the hydrodynamic,

---
[11] A $Mflops/s$ refers to millions of floating point operations per second.

salinity, sediment and water quality models on the IRL coarse grid using a time step of 60 s (a 5 day simulation).

The four scenarios include temporal and spatial boundary conditions for wind stress, tidal elevation at the inlets, precipitation and evaporation, and discharge. The flushing model consists of 9 segments, the sediment model includes both coarse and fine sediments as well as a wave model, and the nutrient model includes the nitrogen and phosphorus cycles, algae, zooplankton and light attenuation.

On the three computing platforms, each of the four scenarios is simulated using both the serial and parallel CH3D models. The parallel model is executed using from one to the total number of processors on a given platform. Each procedure is timed separately in order to asses the effectiveness of each parallel section of the model[12]. Tables showing the time per model iteration for all of the combinations of computing platforms and scenarios are presented in Appendix N.

After each parallel simulation completed, the output was compared to the corresponding serial output to determine the accuracy of the parallel model. In all cases the output was identical[13].

The serial CPU times for the four scenarios on the three computing environments, "ocean", "seiche" and "nereus" average about 0.47, 0.74, 1.83 and 0.42 s per time step. The computing platform "nereus" is approximately 9% faster than the average, "ocean" is approximately 6% slower than the average and "seiche" is approximately equal to the average. It is noted that even though "ocean" is the "fastest"

---

[12]By timing each procedure individually some computational overhead is induced; however, its effect on the overall simulation time is negligible.

[13]Several variables used to check conservation of the sediment and water quality routines differed slightly in their last significant digit on sporadic occasions. In the serial code, these variables are summed over every grid cell in order, while in the parallel model, these variables are summed over every grid cell on a given processor and then totaled using a parallel reduction. The discrepancy in the values is caused by rounding error associated with summing the values in a different order. These discrepancies have no impact on any other facet of the model.

computer as measured by the Linpack Benchmark, it is the "slowest" in the practical applications of the various CH3D models.

For the fine grid simulations, the hydrodynamic, salinity, sediment and water quality simulation, Scenario #3, takes approximately twice as long as hydrodynamic, salinity and flushing simulation, Scenario #2, which takes approximately twice as long as the hydrodynamic and salinity simulation, Scenario #1. A fourfold increase is CPU time occurs when sediment and water quality are added to the hydrodynamic and salinity simulations on the fine grid. This increase coupled with the fourfold decrease in the total number of grid cells in the coarse grid explains why the CPU time for the fine grid hydrodynamic and salinity simulation, Scenario #1, is approximately the same as the coarse grid hydrodynamic, salinity, sediment and water quality simulation, Scenario #4.

A one-year simulation (365 days) for each of the four scenarios, assuming time steps of 60 s (525600 iterations) for Scenarios #1, #2, and #4 and a 30 s (1051200 iterations) time step for Scenario #3, would theoretically take 2.9, 4.5, 22.3 and 2.6 days of computation time, respectively[14]. Using three weeks of computation time, numerous fine grid IRL water quality model simulations become impractical which is why the coarse grid was originally developed. By approximately halving the number of grid cells in the x- and y-directions, the total number of cells is reduced by a factor of four and the grid spacing is doubled. With a larger grid spacing, the time step can be doubled, combined with the 4x reduction in the number of grid cells, lead to an approximate 8x reduction in computation time. However, because accuracy is lost when the resolution of the grid is decreased, the coarse grid is used primarily for the initial water quality model calibration and validation with the final production simulations performed on the fine grid.

---

[14]The theoretical times for Scenarios #1, #2 and #4 have been verified; however, Scenario #3 has not.

Computational speedup is defined as the ratio of the execution time of the serial code to the execution time of the parallel code. The maximum speedup is governed by Amdahl's Law (Amdahl, 1967), $S_w = \frac{1}{f_s + \frac{f_p}{n}}$, where S_w is the maximum theoretical wall clock speedup, f_s is the % of serial work, f_p is the % of parallel work $(f_s + f_p = 1)$ and n is the number of processors. For a 100% parallel code, the maximum speedup equals the number of processors. Table 4.34 illustrates the maximum theoretical speedup for several different cases and shows how little speedup improvement is possible for all but highly parallel codes.

Table 4.34: Theoretical limits of parallel processing performance (Amdahl's Law of maximum speedup). n is the number of processors used.

f_p	$n = 2$	$n = 3$	$n = 4$	$n = 8$	$n = 32$	$n = 256$	$n = 1024$
0.700	1.54	1.88	2.11	2.58	3.17	3.30	3.36
0.800	1.67	2.14	2.50	3.33	4.44	4.92	4.99
0.900	1.82	2.50	3.08	4.71	7.81	9.66	9.91
0.950	1.91	2.73	3.48	5.93	12.55	18.62	19.63
0.970	1.94	2.83	3.67	6.61	16.58	29.60	32.31
0.990	1.98	2.94	3.88	7.48	24.43	72.11	91.18
0.999	2.00	2.99	3.99	7.94	31.04	203.98	506.18

The parallel speedup for the CH3D IRL fine grid scenarios described earlier (Scenarios #1, #2 and #3) on all three computing platforms is shown in Figures 4.26 and 4.27. As the lines get close to the theoretical maximum, the parallel model is performing better. The exact difference between the lines and the theoretical maximum is known as efficiency and is discussed in the next section. These figures show that the combined hydrodynamic, salinity, sediment and water quality simulation showed the best speedup on all the computing platforms. Among the different computing environments, "nereus" showed the highest speedup for a given scenario; however, the computationally more intensive Scenario #3 had nearly identical speedups for

each of the computing platforms (up to two processors.) The curves also indicate a linear behavior of the speedup, up to the maximum of four processors for all scenarios and computing platforms. A comparison between the IRL hydrodynamic, salinity, sediment and water quality fine grid simulation (Scenario #3) and the coarse grid simulation (Scenario #4) is shown in Figure 4.28. This figure illustrates how a more computationally intensive simulation has a much higher speedup. For example, using two processors, the speedup increases by 12%, 9% and 4% on the three computing platforms, respectively. The computing environment which had the largest time per iteration (or the "slowest" real time), "ocean", had the greatest improvement in speedup when comparing the coarse grid simulation to the fine grid simulation. The speedups of the individual parallel procedures are shown as values inside parenthesis in the tables of Appendix N and also illustrate how computationally intensive routines have the higher speedups.

The additional code needed to make sections of the code parallel add time consuming overhead to the model which is not present in the serial version. This can be illustrated on the less computationally intense hydrodynamic and salinity simulation, Scenario #1. On the three computing platforms, "ocean", "seiche" and "nereus", the $n = 1$ parallel CH3D code is approximately 4% slower, 3% slower and the same as the serial CH3D code, respectively. For more computationally intense simulations, the amount of overhead versus the amount of productive work decreases which in turn lowers the reductions of simulation time. For example, $n = 1$ parallel simulation of the more computationally intensive fine grid water quality simulation, Scenario #3, is only 2% slower, 1% slower and the same as the serial CH3D code, respectively. The discrepancies in the percentages among the different computing platforms illustrates that the effect of the overhead is also machine dependent.

As was mentioned earlier, Amdahl's Law (Amdahl, 1967) governs the maximum speedup. Since this law is a function of the percentages of serial and parallel
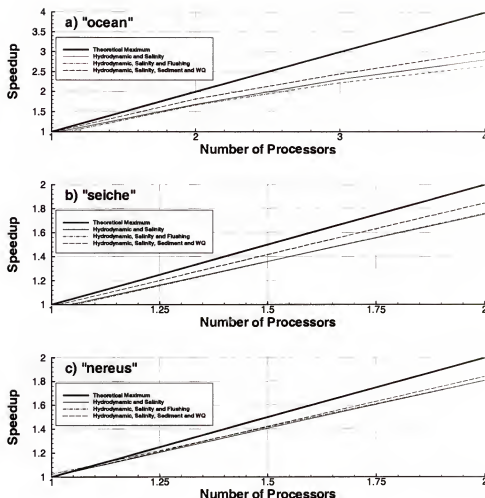
Figure 4.26: Parallel speedup of the fine grid IRL simulation gained through parallelization of the various CH3D models on the 3 computing platforms.

work, the maximum achievable speedup can be calculated for the parallel CH3D model. To this end, the timing results were analyzed to given the percentage of time spent in parallel code, Table 4.35. This table shows that 94.7% of the CPU time for the hydrodynamic and salinity (Scenario #1) and the hydrodynamic, salinity and flushing (Scenario #2) simulations are spent in the parallel procedures while 99.6% of the CPU time for the hydrodynamic, salinity, sediment and water quality simulations (Scenario #3) is spent in the parallel procedures. These data also show how as the number of processors increases the amount of time spent in parallel procedures decreases causing the speedup to deviate from its theoretical maximum. For example,
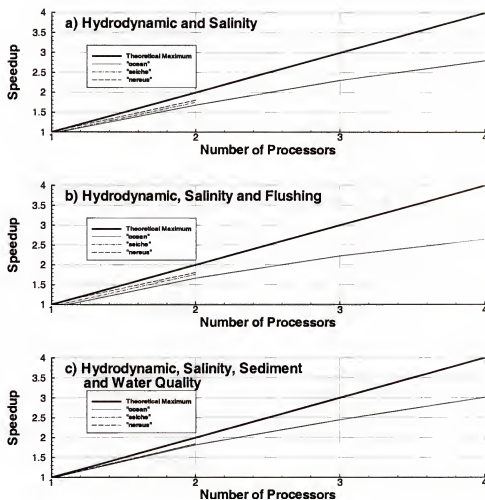
Figure 4.27: Parallel speedup of the various CH3D models during a fine grid IRL simulation.

the hydrodynamic and salinity simulation (Scenario #1) on "ocean" shows that for each of the four processor configurations only 95%, 91%, 88% and 85% of CPU time is spent in parallel procedures. Looking at the appropriate speedup line on Figure 4.26, it becomes evident that the hydrodynamic and salinity simulation (Scenario #1) speedup curve is the least linear of all of the four processor simulations. If the number of processors were to be increased, this curve would asymptotically approach a value much sooner than the more computationally intensive scenarios.

In general, it is difficult to convert all of the CH3D serial code to parallel code. File input and output becomes much more complicated when attempted in
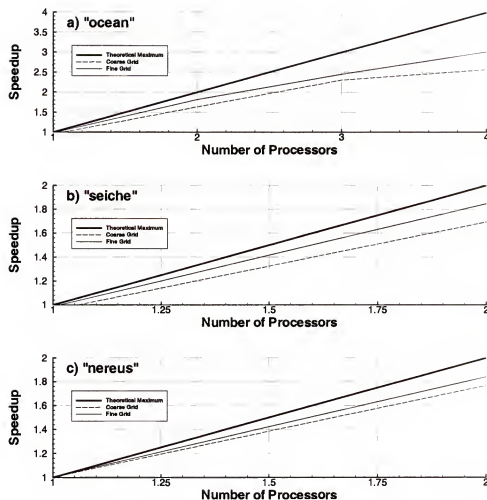
Figure 4.28: A comparison between parallel speedup of the coarse and fine grid IRL simulations of hydrodynamics, salinity, sediment and water quality gained through parallelization of the various CH3D models on the 3 computing platforms.

parallel. In addition, several sections which have small loop sizes or a small amount of computation during the loop were not made parallel. Any performance gain in parallelizing these loops would be lost in overhead necessary to perform these loops in parallel.

Using Amdahl's formulation and an average percentage spent in parallel routines, it is possible to calculate a maximum achievable speedup for the parallel CH3D model. Table 4.36 calculates the theoretical limits of parallel processing performance using the actual percentage of parallel code in the parallel CH3D model.

Table 4.35: Percentage of time spent in parallelized routines. The abbreviations: H, Sa, F, Sed and WQ stand for hydrodynamic, salinity, flushing, sediment and water quality, respectively. Results shown are for simulations using the boundary-fitted fine grid.

Simulation Type	Platform	Serial	$n=1$	$n=2$	$n=3$	$n=4$
	"ocean"	94.8	95.0	91.0	87.5	84.9
H and Sa	"seiche"	94.1	94.4	90.2		
	"nereus"	95.3	95.3	91.5		
	"ocean"	95.4	96.0	92.3	89.7	87.7
H, Sa and F	"seiche"	95.0	95.2	91.6		
	"nereus"	96.5	96.5	93.8		
	"ocean"	99.6	99.6	98.5	97.6	96.7
H, Sa, Sed and WQ	"seiche"	99.4	99.4	98.4		
	"nereus"	99.9	99.9	98.9		

Table 4.36: Theoretical limits of parallel processing performance for the Indian River Lagoon fine grid simulations. (Amdahl's Law of maximum speedup). n is the number of processors used.

Simulation Type	Platform	f_p	$n=2$	$n=3$	$n=4$	$n=8$	$n=32$	$n=256$
	"ocean"	0.948	1.90	2.72	3.46	5.87	12.25	17.95
H and Sa	"seiche"	0.941	1.89	2.68	3.40	5.66	11.31	15.96
	"nereus"	0.953	1.91	2.74	3.51	6.02	13.02	19.72
	"ocean"	0.954	1.91	2.75	3.51	6.05	13.19	20.11
H, Sa and F	"seiche"	0.950	1.90	2.73	3.48	5.93	12.55	18.62
	"nereus"	0.965	1.93	2.80	3.62	6.43	15.35	25.79
	"ocean"	0.996	1.99	2.98	3.95	7.78	28.47	126.73
H, Sa, Sed and WQ	"seiche"	0.994	1.99	2.96	3.93	7.68	26.98	101.19
	"nereus"	0.999	2.00	2.99	3.99	7.94	31.04	203.98

Differences between the theoretically calculated speedups and actual measured speedups could have several causes:

- 1) The main time stepping loop is inherently serial.

- 2) Overhead is induced when using parallel constructs.

- 3) Not all code in a given timed procedure is parallel.

- 4) There is overhead associated with calls to procedures inside of parallelized loops.

- 5) A real "shared memory" architecture does not exist, so additional time is spent acquiring and releasing memory locks.

Most of these problems are not easily remedied; however, at least one type of problem can be. The calls to individual subroutines inside parallel loops could be replaced by the actual code of the called subroutine; thus, removing the overhead associated with the subroutine calls.

4.5.4 Efficiency

Another measure of parallel algorithm performance is efficiency (the fraction of time that the processors spend performing calculations). It is calculated by dividing the speedup by the maximum theoretical speedup (the number of processors used for computation). Table 4.37 lists the efficiencies for the hydrodynamic and salinity, hydrodynamic, salinity and flushing, and the hydrodynamic, salinity, sediment and water quality models on the three computing platforms for the fine grid IRL simulations. The most efficient computing platform is "nereus" while the least efficient is "ocean". For a given platform, the hydrodynamic, salinity, sediment and water quality simulation is the most efficient.

While efficiency illustrates how effective the parallelization of the whole CH3D model is, it is also important to see how effective the parallelization is by itself, this quantity is called relative efficiency. Efficiency divides by the maximum speedup (the number of processors) while relative efficiency divides by the maximum theoretical speedup as calculated by Amdahl's Law (Amdahl, 1967). The relative efficiency; thus, separates out the inherently serial procedures and quantifies how well the actual encoded parallelization does. Table 4.38 lists the relative efficiencies for the hydrodynamic and salinity, hydrodynamic, salinity and flushing, and the hydrodynamic,

Table 4.37: Parallel speedup efficiencies. The abbreviations: H, Sa, F, Sed and WQ stand for hydrodynamic, salinity, flushing, sediment and water quality, respectively. Results shown are using the boundary-fitted fine grid.

Simulation Type	Platform	Serial	$n = 1$	$n = 2$	$n = 3$	$n = 4$
	"ocean"	100.0	96.1	84.1	77.0	70.1
H and Sa	"seiche"	100.0	96.6	87.7		
	"nereus"	100.0	100.4	90.4		
	"ocean"	100.0	91.7	83.3	74.2	66.1
H, Sa and F	"seiche"	100.0	95.7	88.0		
	"nereus"	100.0	102.4	90.5		
	"ocean"	100.0	98.4	90.8	81.6	75.3
H, Sa, Sed and WQ	"seiche"	100.0	98.6	92.3		
	"nereus"	100.0	100.3	92.1		

Table 4.38: Relative parallel percentage efficiencies as calculated using the maximum speedup given by Amdahl's Law (Amdahl, 1967). The abbreviations: H, Sa, F, Sed and WQ stand for hydrodynamic, salinity, flushing, sediment and water quality, respectively.

Simulation Type	Platform	Serial	$n = 1$	$n = 2$	$n = 3$	$n = 4$
	"ocean"	100.0	96.1	88.5	85.0	81.1
H and Sa	"seiche"	100.0	96.6	92.8		
	"nereus"	100.0	100.4	94.6		
	"ocean"	100.0	91.7	87.1	80.9	75.1
H, Sa and F	"seiche"	100.0	95.7	92.5		
	"nereus"	100.0	102.4	93.6		
	"ocean"	100.0	98.4	91.2	82.3	76.3
H, Sa, Sed and WQ	"seiche"	100.0	98.6	92.9		
	"nereus"	100.0	100.3	92.1		

salinity, sediment and water quality models on the three computing platforms for the fine grid IRL simulations. Since these relative efficiencies are not dramatically higher than the standard efficiencies, it is concluded that the procedures may be further parallelized.

CHAPTER 5
THE PARALLEL ENVIRONMENTAL MODEL (PEM)

This chapter discusses the governing equations and solution algorithms used in the newly developed PEM. The PEM is a 2-D vertically averaged, wetting and drying numerical model which is used to simulate the flooding caused by large coastal storms. The PEM has explicit, semi-implicit and fully implicit modes for the propagation, bottom friction and Coriolis terms. The Eulerian-Lagrangian Method is used so solve the advection and diffusion terms and a simple exponential decay of atmospheric pressure formulation is used for the storm model. The model was developed specifically for use on parallel computer systems and will function accordingly in either explicit of implicit modes.

5.1 Governing Hydrodynamic Equations

The governing three-dimensional Cartesian equations describing constant density, free surface flow can be derived from the Navier-Stokes equations were shown previously (Equations 2.1, 2.2, and 2.3).

5.2 Vertical Boundary Conditions

The boundary conditions at the free surface are

$$
\begin{aligned}
A_V \frac{\partial u}{\partial z} &= \frac{\tau_x^w}{\rho} \\
A_V \frac{\partial v}{\partial z} &= \frac{\tau_y^w}{\rho}
\end{aligned}
\tag{5.1}
$$

The wind stress is determined from the wind velocity using

$$
\begin{aligned}
\tau_x^w &= \rho_a C_{ds} u_w \sqrt{u_w^2 + v_w^2} \\
\tau_y^w &= \rho_a C_{ds} v_w \sqrt{u_w^2 + v_w^2}
\end{aligned}
\tag{5.2}
$$

where τ_x^w and τ_y^w are the components of the wind stress, ρ_a is the density of air
(1.293 kg/m^3), u_w and v_w are the components of the wind speed measured at some
height about the water surface and C_{ds} is the wind speed drag coefficient.

Many possible formulations for the drag coefficient are available. Table 5.1
illustrates several formulations popular with storm surge models. As can be seen
in Figure 5.1, the formulations produce fairly linear curves for wind speeds above
10 m/s, with the exception of the Van Dorn (1953) formulation which is significantly
less at higher wind speeds. The two cases of the Hsu (1995) formulation, which
takes into account surface roughness through significant wave height and period, are
for Hurricane Kate (1985) in the Gulf of Mexico and Hurricane Gloria (1985) in
the Atlantic Ocean, respectively. However, while this formulation is more rigorous, it
does not produce markedly different curves than the much simpler linear formulations
of Smith and Banke (1975) and Garratt (1977). For simplicity and the lack of a wave
model to give estimates of wave height and period, the Garratt (1977) formulation
was chosen as the drag formulation used in this model.

The boundary conditions at the bottom are

$$A_V \frac{\partial u}{\partial z} = \frac{\tau_x^b}{\rho} \\ A_V \frac{\partial v}{\partial z} = \frac{\tau_y^b}{\rho} \tag{5.3}$$

The bottom stress is given using

$$\tau_x^b = \frac{\rho g U |V|}{C_z^2} \\ \tau_y^b = \frac{\rho g V |V|}{C_z^2} \tag{5.4}$$

where the Chezy coefficient, C_z is given by

$$C_z = 1.0 \times \frac{R^{\frac{1}{6}}}{n} \tag{5.5}$$

and the hydraulic radius R is given in meters and n is Manning's n. In shallow
estuaries, the hydraulic radius is approximated by the total depth.

Table 5.1: Descriptions of popular wind stress drag coefficients applied to storm surge models. W_s is the magnitude of wind speed (m/s) measured at 10 meters above the water surface, ρ_w and ρ_a are the densities of water and air, respectively, $k_1 = 1.1 \times 10^{-6}$, $k_2 = 2.5 \times 10^{-6}$, $W_{cr} = 7.2 \ m/s$ (14 knots), T_p is the wave period at the spectral peak, H_s is the significant wave height and g is the gravitational acceleration.

Author(s)	Formula for the wind stress drag coefficient, C_{ds}
Van Dorn (1953)	$\frac{\rho_w}{\rho_a} \times \begin{cases} k_1, & \text{if } W_s < W_{cr} \\ k_1 + k_2\left(1 - \frac{W_{cr}}{W_s}\right)^2, & \text{if } W_s \geq W_{cr} \end{cases}$
Smith and Banke (1975)	$0.001 \times (0.63 + 0.066 W_s)$
Garratt (1977)	$0.001 \times (0.75 + 0.067 W_s)$
Hsu (1995)	$\left[\dfrac{0.4}{11.0 - \ln\left(\dfrac{H_s}{\left(\frac{gT_p}{2\pi W_s}\right)^{2.6}} \right)} \right]^2$

5.3 Differential Equations for the Parallel Environmental Model

Vertically averaging the continuity and momentum equations over the depth and applying the vertical boundary conditions yield the following equations for continuity and x- and y-momentum

$$\frac{\partial \zeta}{\partial t} + \frac{\partial}{\partial x}(UH) + \frac{\partial}{\partial y}(VH) = 0 \tag{5.6}$$

$$\frac{\partial U}{\partial t} + U\frac{\partial U}{\partial x} + V\frac{\partial U}{\partial y} = -g\frac{\partial \zeta}{\partial x} + A_H\left(\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2}\right) - \frac{C_d U|V|}{H} + fV + \frac{\tau_x^w}{\rho H} \tag{5.7}$$

$$\frac{\partial V}{\partial t} + U\frac{\partial V}{\partial x} + V\frac{\partial V}{\partial y} = -g\frac{\partial \zeta}{\partial y} + A_H\left(\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2}\right) - \frac{C_d V|V|}{H} - fU + \frac{\tau_y^w}{\rho H} \tag{5.8}$$

where H is the total depth $(h + \zeta)$, U and V are the depth averaged velocities $\frac{1}{H}\int_{-h}^{\zeta} u \ dz$ and $\frac{1}{H}\int_{-h}^{\zeta} v \ dz$, respectively, and C_d is a dimensional drag coefficient defined as $\frac{g}{C_z^2}$.
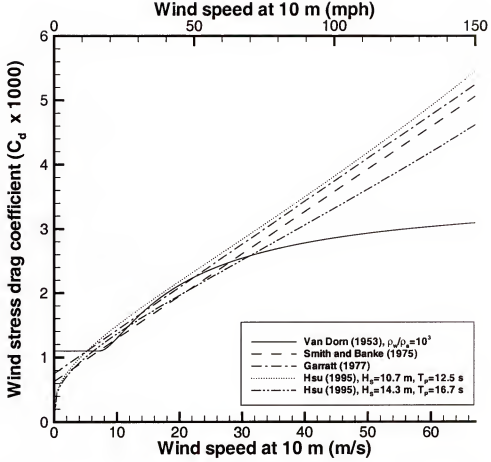
Figure 5.1: Calculated wind stress drag coefficients using several popular formulations.

<u>5.4 Finite Difference Equations for the Explicit Model</u>

The governing differential equations for the continuity, x- and y-momentum given in Equations 5.6, 5.7, and 5.8 are first given an explicit discretization of the propagation and bottom friction terms. These discretization take the form

$$\frac{\zeta_{i,j}^{n+1} - \zeta_{i,j}^{n}}{\Delta t} + \frac{\left(U_{i+1,j}^{n}H_{u,i+1,j}^{n} - U_{i,j}^{n}H_{u,i,j}^{n}\right)}{\Delta x} + \frac{\left(V_{i,j+1}^{n}H_{v,i,j+1}^{n} - V_{i,j}^{n}H_{v,i,j}^{n}\right)}{\Delta y} = 0 \quad (5.9)$$

$$\frac{U_{i,j}^{n+1} - U_{i,j}^{n}}{\Delta t} = -\frac{g}{\Delta x}\left(\zeta_{i,j}^{n} - \zeta_{i-1,j}^{n}\right) - \frac{C_d}{H_{u,i,j}^{n}}|V|_{u,i,j}^{n}U_{i,j}^{n} + \frac{\tau_{x,i,j}^{w}}{\rho H_{u,i,j}^{n}} + F_{u,i,j}^{n} \quad (5.10)$$

$$\frac{V_{i,j}^{n+1} - V_{i,j}^{n}}{\Delta t} = -\frac{g}{\Delta y}\left(\zeta_{i,j}^{n} - \zeta_{i,j-1}^{n}\right) - \frac{C_d}{H_{v,i,j}^{n}}|V|_{v,i,j}^{n}V_{i,j}^{n} + \frac{\tau_{y,i,j}^{w}}{\rho H_{v,i,j}^{n}} + F_{v,i,j}^{n} \quad (5.11)$$

where $F_{u,i,j}^n$ and $F_{v,i,j}^n$ represent the remaining non-linear, diffusion and pressure gradient terms in the x- and y-directions.

5.5 Finite Difference Equations for the Semi-Implicit Model

The governing differential equations for the continuity, x- and y- momentum given in Equations 5.6, 5.7, and 5.8 are given a semi-implicit discretization of the propagation and bottom friction terms. The Coriolis term is also given a semi-implicit discretization; however, only in the y-momentum equation. These discretization take the form

$$
\begin{aligned}
\frac{\zeta_{i,j}^{n+1} - \zeta_{i,j}^n}{\Delta t} \quad & + \quad \frac{\theta_1}{\Delta x}\left(U_{i+1,j}^{n+1}H_{u,i+1,j}^n - U_{i,j}^{n+1}H_{u,i,j}^n\right) \\
& + \quad \frac{\theta_1}{\Delta y}\left(V_{i,j+1}^{n+1}H_{v,i,j+1}^n - V_{i,j}^{n+1}H_{v,i,j}^n\right) \\
& + \quad \frac{1-\theta_1}{\Delta x}\left(U_{i+1,j}^{n}H_{u,i+1,j}^n - U_{i,j}^{n}H_{u,i,j}^n\right) \\
& + \quad \frac{1-\theta_1}{\Delta y}\left(V_{i,j+1}^{n}H_{v,i,j+1}^n - V_{i,j}^{n}H_{v,i,j}^n\right) = 0
\end{aligned}
\tag{5.12}
$$

$$
\begin{aligned}
\frac{U_{i,j}^{n+1} - U_{i,j}^n}{\Delta t} = \quad & - \quad \frac{g\theta_1}{\Delta x}\left(\zeta_{i,j}^{n+1} - \zeta_{i-1,j}^{n+1}\right) - \frac{g(1-\theta_1)}{\Delta x}\left(\zeta_{i,j}^n - \zeta_{i-1,j}^n\right) \\
& - \quad \frac{C_d}{H_{u,i,j}^n}|V|_{u,i,j}^n U_{i,j}^{n+1}\theta_2 - \frac{C_d}{H_{u,i,j}^n}|V|_{u,i,j}^n U_{i,j}^n(1-\theta_2) \\
& + \quad fV_{u,i,j}^n + \frac{\tau_{x,i,j}^w}{\rho H_{u,i,j}^n} + F_{u,i,j}^n
\end{aligned}
\tag{5.13}
$$

$$
\begin{aligned}
\frac{V_{i,j}^{n+1} - V_{i,j}^n}{\Delta t} = \quad & - \quad \frac{g\theta_1}{\Delta y}\left(\zeta_{i,j}^n - \zeta_{i,j-1}^{n+1}\right) - \frac{g(1-\theta_1)}{\Delta y}\left(\zeta_{i,j}^n - \zeta_{i,j-1}^n\right) \\
& - \quad \frac{C_d}{H_{v,i,j}^n}|V|_{v,i,j}^n V_{i,j}^{n+1}\theta_2 - \frac{C_d}{H_{v,i,j}^n}|V|_{v,i,j}^n V_{i,j}^n(1-\theta_2) \\
& - \quad fU_{v,i,j}^{n+1}\theta_3 - fU_{v,i,j}^n(1-\theta_3) + \frac{\tau_{y,i,j}^w}{\rho H_{v,i,j}^n} + F_{v,i,j}^n
\end{aligned}
\tag{5.14}
$$

where $F_{u,i,j}^n$ and $F_{v,i,j}^n$ represent the remaining non-linear, diffusion and pressure gradient terms in the x- and y-directions, respectively and θ_1, θ_2, and θ_3 are the degrees of implicitness of the surface slope, bottom friction and Coriolis terms, respectively.

The x-momentum equation (5.13) is first solved for $U_{i,j}^{n+1}$ and then substituted into the Coriolis term in the y-momentum equation (5.14) using the following equation

$$
-fU_{v,i,j}^{n+1}\theta_3 = -f\frac{\left(U_{i,j}^{n+1} + U_{i+1,j}^{n+1} + U_{i,j-1}^{n+1} + U_{i+1,j-1}^{n+1}\right)}{4}\theta_3
\tag{5.15}
$$

The new y-momentum finite difference equation is then solved for $V_{i,j}^{n+1}$. The i's in the $U_{i,j}^{n+1}$ equation and the j's in the $V_{i,j}^{n+1}$ are incremented by one yielding $U_{i+1,j}^{n+1}$ and $V_{i,j+1}^{n+1}$, respectively. The resulting 4 momentum finite difference equations,

$U_{i,j}^{n+1}$, $U_{i+1,j}^{n+1}$, $V_{i,j}^{n+1}$, and $V_{i,j+1}^{n+1}$, are substituted back into the continuity equation (5.12) resulting in the following 9-diagonal system of linear equations for the surface elevation, $\zeta_{i,j}^{n+1}$.

$$
\begin{matrix}
\Pi_{nw}\zeta_{i-1,j+1}^{n+1} & \Pi_n\zeta_{i,j+1}^{n+1} & \Pi_{ne}\zeta_{i+1,j+1}^{n+1} \\
\Pi_w\zeta_{i-1,j}^{n+1} & \Pi_c\zeta_{i,j}^{n+1} & \Pi_e\zeta_{i+1,j}^{n+1} \\
\Pi_{sw}\zeta_{i-1,j-1}^{n+1} & \Pi_s\zeta_{i,j-1}^{n+1} & \Pi_{se}\zeta_{i+1,j-1}^{n+1}
\end{matrix}
= (RHS)_{i,j}^n \tag{5.16}
$$

where

$$
\begin{aligned}
\Pi_{nw} &= -\tfrac{\Gamma_{i,j+1}}{4}\tfrac{g\Delta t}{\Delta x}\theta_1\beta_{u,i,j+1} \\
\Pi_{ne} &= +\tfrac{\Gamma_{i,j+1}}{4}\tfrac{g\Delta t}{\Delta x}\theta_1\beta_{u,i+1,j+1} \\
\Pi_n &= -\Pi_{nw}-\Pi_{ne}-\Phi_{v,i,j+1} \\
\Pi_{sw} &= +\tfrac{\Gamma_{i,j}}{4}\tfrac{g\Delta t}{\Delta x}\theta_1\beta_{u,i,j-1} \\
\Pi_{se} &= -\tfrac{\Gamma_{i,j}}{4}\tfrac{g\Delta t}{\Delta x}\theta_1\beta_{u,i+1,j-1} \\
\Pi_s &= -\Pi_{sw}-\Pi_{se}-\Phi_{v,i,j} \\
\Pi_w &= +\tfrac{\Gamma_{i,j}}{4}\tfrac{g\Delta t}{\Delta x}\theta_1\beta_{u,i,j} - \tfrac{\Gamma_{i,j+1}}{4}\tfrac{g\Delta t}{\Delta x}\theta_1\beta_{u,i,j} - \Phi_{u,i,j} \\
\Pi_e &= -\tfrac{\Gamma_{i,j}}{4}\tfrac{g\Delta t}{\Delta x}\theta_1\beta_{u,i+1,j} + \tfrac{\Gamma_{i,j+1}}{4}\tfrac{g\Delta t}{\Delta x}\theta_1\beta_{u,i+1,j} - \Phi_{u,i+1,j} \\
\Pi_c &= 1 - \Pi_w - \Pi_e - \Pi_n - \Pi_{nw} - \Pi_{ne} - \Pi_s - \Pi_{sw} - \Pi_{se}
\end{aligned} \tag{5.17}
$$

$$
\begin{aligned}
(RHS)_{i,j}^n = \zeta_{i,j}^n \;\; & + \tfrac{\Delta t}{\Delta x}\theta_1 H_{u,i,j}^n\alpha_{u,i,j}\beta_{u,i,j} \\
& - \tfrac{\Delta t}{\Delta x}\theta_1 H_{u,i+1,j}^n\alpha_{u,i+1,j}\beta_{u,i+1,j} \\
& + \tfrac{\Delta t}{\Delta y}\theta_1 H_{v,i,j}^n\alpha_{v,i,j}\beta_{v,i,j} \\
& - \tfrac{\Delta t}{\Delta y}\theta_1 H_{v,i,j+1}^n\alpha_{v,i,j+1}\beta_{v,i,j+1} \\
& + \tfrac{\Delta t(1-\theta_1)}{\Delta x}\left(U_{i,j}^n H_{u,i,j}^n - U_{i+1,j}^n H_{u,i+1,j}^n\right) \\
& + \tfrac{\Delta t(1-\theta_1)}{\Delta y}\left(V_{i,j}^n H_{v,i,j}^n - V_{i,j+1}^n H_{v,i,j+1}^n\right) \\
& - \tfrac{\Gamma_{i,j}}{4}\left(\begin{array}{ll} \alpha_{u,i,j}\beta_{u,i,j} & +\alpha_{u,i+1,j}\beta_{u,i+1,j} \\ +\alpha_{u,i,j-1}\beta_{u,i,j-1} & +\alpha_{u,i+1,j-1}\beta_{u,i+1,j-1} \end{array}\right) \\
& + \tfrac{\Gamma_{i,j+1}}{4}\left(\begin{array}{ll} \alpha_{u,i,j+1}\beta_{u,i,j+1} & +\alpha_{u,i+1,j+1}\beta_{u,i+1,j+1} \\ +\alpha_{u,i,j}\beta_{u,i,j} & +\alpha_{u,i+1,j}\beta_{u,i+1,j} \end{array}\right)
\end{aligned} \tag{5.18}
$$

$$
\begin{aligned}
\alpha_{u,i,j} = \;\; & \Delta t F_{u,i,j}^n + U_{i,j}^n \\
& + f V_{u,i,j}^n \Delta t \\
& - \tfrac{g\Delta t}{\Delta x}\left(\zeta_{i,j}^n - \zeta_{i-1,j}^n\right)(1-\theta_1) \\
& - \tfrac{C_d \Delta t}{H_{u,i,j}^n}|V|_{u,i,j}^n U_{i,j}^n (1-\theta_2) + \tfrac{\Delta \tau_{x,i,j}^w}{\rho H_{u,i,j}^n} + \Delta t F_{u,i,j}^n
\end{aligned} \tag{5.19}
$$

$$
\begin{aligned}
\alpha_{v,i,j} = \;\; & \Delta t F_{v,i,j}^n + V_{i,j}^n \\
& - f U_{v,i,j}^n \Delta t(1-\theta_3) \\
& - \tfrac{g\Delta t}{\Delta y}\left(\zeta_{i,j}^n - \zeta_{i,j-1}^n\right)(1-\theta_1) \\
& - \tfrac{C_d \Delta t}{H_{v,i,j}^n}|V|_{v,i,j}^n V_{i,j}^n (1-\theta_2) + \tfrac{\Delta \tau_{y,i,j}^w}{\rho H_{v,i,j}^n} + \Delta t F_{v,i,j}^n
\end{aligned} \tag{5.20}
$$

$$
\beta_{u,i,j} = \frac{1}{1 + \tfrac{C_d \Delta t}{H_{u,i,j}^n}|V|_{u,i,j}^n\theta_2} \tag{5.21}
$$

$$\beta_{v,i,j} = \frac{1}{1 + \frac{C_d \Delta t}{H^n_{v,i,j}} |V|^n_{v,i,j} \theta_2} \tag{5.22}$$

$$\Gamma_{i,j} = f \theta_1 \theta_3 \frac{\Delta t}{\Delta y} H^n_{v,i,j} \beta_{v,i,j} \Delta t \tag{5.23}$$

$$\Phi_{u,i,j} = \frac{\Delta t}{\Delta x} \theta_1 H^n_{u,i,j} \frac{g \Delta t}{\Delta x} \theta_1 \beta_{u,i,j} \tag{5.24}$$

$$\Phi_{v,i,j} = \frac{\Delta t}{\Delta y} \theta_1 H^n_{v,i,j} \frac{g \Delta t}{\Delta y} \theta_1 \beta_{v,i,j} \tag{5.25}$$

5.6 The Parallel Eulerian-Lagrangian Method (ELM)

A major difficulty in numerical modeling of free surface flows is the accurate and efficient treatment of the advection and diffusion terms in the governing equations. Numerous different numerical schemes are available to solve these terms; however, few are unconditionally stable and most have rigid time step limitations. Since the intended use of the PEM is in large scale applications with a large number of grid cells, it is important that the model uses time steps as large as possible to minimize simulation run time. Hence, the optimal method for the PEM model should be unconditionally stable. To this end, the ELM is used in the PEM to solve the advection and diffusion terms in the governing equations.

The ELM uses the Lagrangian form of the governing equations in an Eulerian computational grid system. The method is generally referred to as the semi-Lagrangian method in the numerical weather prediction community (Staniforth and Cote, 1991), but is called the Eulerian-Lagrangian method by estuarine and coastal modelers (*e.g.* Casulli and Cheng (1992); Oliveira et al. (2000); Lee (2000)). A previous study did develop a parallel semi-Lagrangian advective scheme for use with an adaptive grid system (Behrens, 1996); however, it was designed for used on a KSR-1 (Kendall Square Research), a virtual shared memory computer, and would not work well on a distributed memory system.

Following in the manner of Casulli and Cheng (1992), the portion of F_u representing the remaining advection and diffusion terms in the x-direction can be discretized using the ELM method resulting in (a similar equation can be developed for F_v, the remaining advection and diffusion terms in the y-direction)

$$
\begin{aligned}
F_{u,i,j}^n = \ & u_{i-a,j-b}^n \\
& + A_H \Delta t \left(\frac{u_{i-a-1,j-b}^n - 2u_{i-a,j-b}^n + u_{i-a+1,j-b}^n}{\Delta x^2} \right) \\
& + A_H \Delta t \left(\frac{u_{i-a,j-b-1}^n - 2u_{i-a,j-b}^n + u_{i-a,j-b+1}^n}{\Delta y^2} \right)
\end{aligned}
\tag{5.26}
$$

where $a = u\Delta t/\Delta x$ and $b = v\Delta t/\Delta y$ are the grid Courant numbers.

Because of the non-linearity of the advective terms, the determination of a and b requires the integration of the streak lines, $dx/dt = u$ and $dy/dt = v$, in which case the right hand sides are know only at time level n. Thus, u and v are assumed to be invariant over a time step and the streak lines will be integrated numerically backwards from time level $n+1$ to n using the Euler method, since the streak lines, which in general are not straight lines, are better approximated.

In general, a and b are not integers; therefore $(i-a, j-b)$ is not a grid point and an interpolation formula must be used to solve for $u_{i-a,j-b}^n$. For positive a and b, let l and m be the integer parts and p and q their corresponding decimal parts, so that $a = l + p$ and $b = m + q$. Then $u_{i-a,j-b}^n$ can be approximated as

$$
\begin{aligned}
u_{i-a,j-b}^n = \ & (1-p) \left[(1-q)u_{i-l,j-m}^n + qu_{i-l,j-m-1}^n \right] \\
& + p \left[(1-q)u_{i-l-1,j-m}^n + qu_{i-l-1,j-m-1}^n \right]
\end{aligned}
\tag{5.27}
$$

In the PEM, each processor controls only one small portion of the overall grid system. For each u and v node on a given processor, a streak line is tracked backwards. Depending on the direction of the flow and the proximity of the node to a processor boundary, the streak lines may be tracked to another processor. When this occurs, the original processor which started back-tracking the streak line does not have enough information to proceed. To handle these type of situations a parallel ELM algorithm was developed.

The parallel ELM algorithm consists of two parts. The first part attempts to trace back every needed streak line within a processors domain (Listing 5.1). If the streak line is tracked to another processor, a message is sent to the appropriate neighbor processor giving it enough information about the current progress of the back-tracked streak line such that the neighbor processor can finish the backward integration. Once this message has been sent, a counter identifying the number of uncompleted streak lines is incremented (NUM_OUTSIDE_GRID) and the processor continues to track backwards the remaining nodes streak lines. Once all streak lines have been tracked backwards or messages sent to neighbor processors telling them to complete the streak line, the first part of the algorithm is complete.

In the second part of the algorithm (Listing 5.2), each processor spin in a loop which 1) checks for completed streak lines sent back from other processors (Line 63-67), 2) checks to see if any neighbor processors have requested that they complete one of their streak lines (Lines 73-95), and 3) check to see if all of its unfinished streak lines have been returned (Lines 101-107). As completed streak lines are returned from neighbor processors, a counter identifying the number of returned completed streak lines is incremented (NUM_MESSAGES_RECEIVED). Once all of a given processors unfinished streak lines have been returned (NUM_MESSAGES_RECEIVED = NUM_OUTSIDE_GRID), a message is sent to all other processors telling them that this processor has completed (Line 105). Upon receipt of a similar message from all other processors, the processors then exit their respective spin loops (Lines 109-121).

Listing 5.1: Pseudocode representation of the parallel ELM algorithm (Part 1/2).

```
1   ! Backtrack (as far as possible) all of the streak lines
2   ! which start in the local subdomain.
3     do J=LOCAL_J_MIN, LOCAL_J_MAX        ! Local subdomain range of J
4       do I=LOCAL_I_MIN, LOCAL_I_MAX      ! Local subdomain range of I
5
6         { Initialize the streak line position.}
7         COUNT=0                           ! Integration steps
8         OTHER_SUB_GRID_FLAG=0             ! Location of streak line
9         NUM_OUTSIDE_GRID=0                ! Number of streak lines outside local grid
10
11  !     Track the streak line backwards a total of NITER steps or if the position
```

```
12   !     of the line goes outside the local grid, send the streak line's position
13   !     to the processor containing that section of the grid.
14         do while ( COUNT < NITER .and. OTHER_SUB_GRID_FLAG == 0 )
15            COUNT++                              ! Increment integration step
16            {Update the streak line position by interpolation of the
17            velocity field at the previous position.}
18            if ( {updated streak line position is outside the local grid} ) then
19               OTHER_SUB_GRID_FLAG=1             ! Set flag to exit integration loop
20               NUM_OUTSIDE_GRID++               ! Increment number of streak lines
21                                                ! outside the local grid
22               {Send a non-blocking message to the processor which contains
23                the streak line at its updated position.  This message
24                contains the updated streak line position, its starting position,
25                the starting processor number and the current integration step}
26            endif
27
28   ! If the backward integration is complete, calculate the variable at
29   ! its final streak position.
30            if ( COUNT == NITER .and. OTHER_SUB_GRID_FLAG == 0 ) then
31               {Calculate the variable at its final streak position.}
32            endif
33
34         end do                                 ! do while ...
35
36       end do                                   ! do I=...
37     end do                                     ! do J=...
```

Listing 5.2: Pseudocode representation of the parallel ELM algorithm (Part 2/2).

```
38   ! Inform other processors when this process has received back all
39   ! the streak lines it sent out
40     INFORM_OTHERS=1
41   ! Number of streak lines sent out that this processor has received back
42     NUM_MESSAGES_RECEIVED=0
43     COMPLETED=.false.                          ! Status of all the processors
44
45   ! Loop continuously until all processors have received all of their streak lines back
46     do while ( COMPLETED == .false. )
47        {Non-blocking check for incoming streak lines.}
48        if ( {Incoming streak line exists} ) then
49
50            {Accept the incoming streak line.}
51
52            if ( {Integration step of incoming streak line == NITER} ) then
53
54               SENT_ON=0                         ! Flag to indicate if a message was sent
55               if ( {Message final destination is not this processor} ) then
56                  {Perform one last backward integration}
57                  {Send a non-blocking message to the processor which contains the streak line at its
58                   updated position.  This message contains the updated streak line position, its
59                   starting position, the starting processor number and the current integration step}
60                  SENT_ON=1
61               endif
62
```

```
63  !     If message was not sent on, its final destination is this processor
64        if ( SENT_ON /= 1 ) then              ! Message was not sent on
65          {Save the variable information passed in the message.}
66          NUM_MESSAGES_RECEIVED++      ! Increment the count of messages received
67                                        ! whose final destination is this processor
68        endif
69
70      else                                    ! Backtrack to get variable
71
72        { Initialize the streak line position to that of the message.}
73  !     Track the streak line backwards a total of NITER steps or if the position
74  !     of the streak line goes outside the local grid, send the streak line's
75  !     position to the processor containing that section of the grid.
76        do while ( COUNT < NITER .and. OTHER_SUB_GRID_FLAG == 0 )
77          COUNT++                            ! Increment integration step
78          {Update the streak line position by interpolation of the
79            velocity field at the previous position.}
80          if ( {updated streak line position is outside the local grid} ) then
81            OTHER_SUB_GRID_FLAG=1    ! Set flag to exit integration loop
82            {Send a non-blocking message to the processor which contains the streak line at its
83              updated position. This message contains the updated streak line position, its
84              starting position, the starting processor number and the current integration step}
85          endif
86        end do                               ! do while ...
87
88  !     If the backward integration is complete, calculate the variable at its final
89  !     streak position and send the result to its processor of origin.
90        if ( COUNT == NITER .and. OTHER_SUB_GRID_FLAG == 0 ) then
91          {Calculate the variable at its final streak position.}
92          {Send a non-blocking message to the processor which contains the streak line at its
93            updated position. This message contains the updated streak line position, its
94            starting position, the starting processor number and the current integration step.}
95        endif
96
97      endif                                  ! Integration step == NITER
98
99    endif                                    ! Incoming streak line
100
101 !   If this processor has received back all of the streak lines it sent to
102 !   other processors, then send a signal to the other processors.
103     if ( NUM_MESSAGES_RECEIVED == NUM_OUTSIDE_GRID .and.
104        INFORM_OTHERS == 1 ) then
105       {Send completion signal to all other processors.}
106       INFORM_OTHERS=0                     ! Only inform others once
107     endif
108
109 !   Check for completion signal from other processors
110     {Non-blocking check for completion signal.}
111     if ( {A completion signal is present} ) then
112       {Read and record which processor sent the signal.}
113     endif
114
115 !   If this processor has received all of its streak lines
116 !   then check to see if all of the other
```

```
117  !   processors have completed their streak lines,
118  !   if so then terminate the main loop.
119      if ( {This processor has received all its streak lines back} ) then
120        if ( {All other processors have received their streaks lines back} ) then
121          COMPLETED=.true.              ! Flag will terminate main loop
122        endif
123      endif
124
125    end do                             ! do while ...
```

5.7 Storm Model Equations

One of the most important parts of a numerical model used to simulate storm surge is the storm model itself. A good storm model is necessary to determine rapidly changing atmospheric pressure gradient and wind stress associated with the passage of a storm. While a planetary boundary layer (PBL) model (Vickery and Twisdale, 1995; Thompson and Cardone, 1996; Vickery et al., 2000) would be the best choice for a storm model, a PBL model is too complex for the purposes of this study. Instead a simple storm model assuming an exponential decay of pressure from the center of a storm is used. The governing pressure gradient terms and wind field (used to calculate the wind stresses, τ_x and τ_y) can be derived as follows.

The local air pressure in a storm, P, can be written as (Holland, 1980)

$$P = P_o + (P_\infty - P_o) e^{-A/r^B} \tag{5.28}$$

where P_o is the pressure in the center on the storm, P_∞ is the freestream pressure, r is the distance from the center of the storm and A and B are scaling parameters. For the purposes of this study, the simpler local air pressure formulation of Wilson (1957) is used (A is set equal to the radius of maximum wind speed, R, and B is set equal to 1). The remaining derivation then follows in the manner of Wilson (1957).

Subtracting P_∞ from both sides and rearranging the right hand side yields,

$$P - P_\infty = (P_o - P_\infty) \left(1 - e^{-R/r}\right) \tag{5.29}$$

The left hand side is the relative atmospheric pressure, P_a, and $P_o - P_\infty$ is the central pressure drop of the storm, ΔP_o. Making these substitutions, the equation can be rewritten as

$$P_a = \Delta P_o \left(1 - e^{-R/r} \right) \tag{5.30}$$

The air pressure term in the x- and y-momentum equations consists of derivatives of P_a with respect to x and y, respectively. The terms, after negating and dividing by water density are

$$-\frac{1}{\rho_w} \frac{\partial P_a}{\partial x} = \frac{\Delta P_o}{\rho_w} \frac{\partial}{\partial x} \left(e^{-R/r} \right) \tag{5.31}$$

$$-\frac{1}{\rho_w} \frac{\partial P_a}{\partial y} = \frac{\Delta P_o}{\rho_w} \frac{\partial}{\partial y} \left(e^{-R/r} \right) \tag{5.32}$$

The finite difference forms of the x- and y-momentum air pressure terms used in the model are evaluated at the u- and v-nodes, respectively

$$-\frac{1}{\rho_w} \frac{\partial P_a}{\partial x} = \frac{\Delta P_o}{\rho_w \Delta x} \left(e^{-R/r_{i,j}} - e^{-R/r_{i-1,j}} \right) \tag{5.33}$$

$$-\frac{1}{\rho_w} \frac{\partial P_a}{\partial y} = \frac{\Delta P_o}{\rho_w \Delta y} \left(e^{-R/r_{i,j}} - e^{-R/r_{i,j-1}} \right) \tag{5.34}$$

The distance, $r_{i,j}$, is measured from the center of the storm to the center of the (i,j) cell.

The storm also influences the elevation at open boundaries, $\zeta_{open} = \zeta_{tide} + \zeta_{storm}$. The surface elevation due to the pressure of the storm can be written as

$$\zeta_{storm} = \frac{P_a}{\rho_w g} = \frac{\Delta P_o}{\rho_w g} \left(1 - e^{-R/r} \right) \tag{5.35}$$

The cyclostrophic wind velocity, U_c, is

$$U_c = \sqrt{-\frac{\Delta P_o}{\rho_a} \frac{R}{r} e^{-R/r}} \tag{5.36}$$

The geostrophic wind velocity, U_g, is

$$U_g = -\frac{\Delta P_o}{f \rho_a} \frac{R}{r^2} e^{-R/r} \tag{5.37}$$

The gradient wind velocity, U_G is

$$U_G = U_c \left(\sqrt{\gamma^2 + 1} - \gamma \right) \tag{5.38}$$

where

$$\gamma = \frac{1}{2} \left(\frac{V_s^*}{U_c} + \frac{U_c}{U_g} \right) \tag{5.39}$$

and the resolved part, V_s^*, of the translational velocity of the storm, V_s is

$$V_s^* = V_s sin\left(\theta \right) \tag{5.40}$$

where θ is the angle from the direction of bearing of the storm, β to any point inside the storm. The surface wind velocity, U_s, in the x- and y-directions is then written as

$$U_{sx} = KU_G cos\left(90 + \theta + \beta + \phi \right) \tag{5.41}$$

$$U_{sy} = KU_G sin\left(90 + \theta + \beta + \phi \right) \tag{5.42}$$

where ϕ is an inward rotation angle of 18° and K is the ratio of surface wind velocity to gradient wind velocity.

5.8   Solution Technique

When performing simulations, PEM operates in either an explicit or semi-implicit mode. Explanation of how the water level and velocities are calculated in these modes follows. For details on how the calculation of water level and velocities relates to the other sub-models see Appendix O. The message passing technique that is used to explicitly pass data between processors conforms to the Message Passing Interface (MPI) standard[1]. Specifically, MPICH[2] (Gropp and Lusk, 2001), a portable version of MPI, libraries are used in the PEM.

---

[1]http://www.unix.mcs.anl.gov/mpi/

[2]http://www.unix.mcs.anl.gov/mpi/mpich/index.html

5.8.1 The Explicit Mode

Due to the stringent time step limitations, the explicit mode is generally used only for testing purposes. First, tidal, wind and pressure boundary conditions are updated for the new time level. Then external mode equations are solved on each processor (Equations 5.9 through 5.11). Next, the boundary water levels and velocities which may be used by a given cell's neighbor processors are passed in the order shown in Figure 5.2. Then, the parallel ELM is used to calculate the non-linear and diffusion terms at the new time level. Finally, the time step is incremented and process of solving the equations begins again.
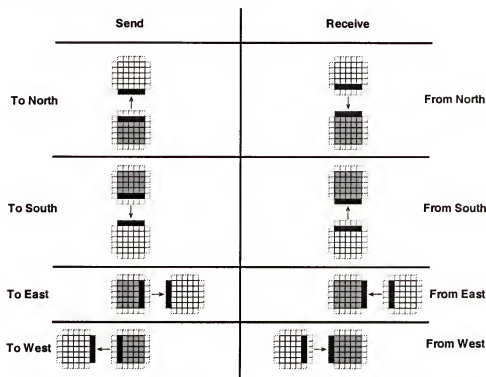
Figure 5.2: The order of the send and receive operations in the PEM. The cells with dotted lines on the receiving processors are termed "ghost cells".

5.8.2 The Semi-implicit Mode

As with the explicit mode, the first step is to update the tidal, wind and pressure gradient boundary conditions for the new time level. Then, the 9-diagonal

system of linear equations for water level (Equation 5.16) are solved using Aztec (Version 2.1) (Tuminaro et al., 1999). Aztec is a parallel iterative library that solves linear systems of equations of the form $Ax = b$, where A is a given $n \times n$ sparse matrix, b is a given vector of length n, and x is the vector of length n to be calculated. Parallel communication in the Aztec routines is based on the MPI standard.

Aztec includes a number of solution and scaling algorithms, preconditioners and residual expressions for determination of convergence; however, not all options are applicable to the 9-diagonal system of equations given in 5.16. The solution algorithms used in this study are conjugate gradient squared (CGS), transpose-free quasi-minimal residual (TFQMR) and bi-conjugate gradient with stabilization (BICGSTAB). The scaling algorithms used are point Jacobi, scaling each row so the magnitude of its elements sum to 1, symmetric scaling so diagonal elements are 1 and symmetric scaling using the matrix row sums. The residual expression chosen to check for convergence is

$$\frac{||r||_2}{||r^{(0)}||_2} \quad < \quad \text{Tolerance} \tag{5.43}$$

Aztec works with two specific sparse matrix formats: 1) a point-entry modified sparse row (MSR) format (Shadid and Tuminaro, 1992) or 2) a block-entry variable block row (VBR) format (Carney et al., 1993). Aztec generalizes these formats for parallel implementation and are referred to as "distributed" yielding DMSR and DVBR, respectively. Further details on Aztec's implementation of these formats can be found in Tuminaro et al. (1999). The PEM uses the MSR format.

After the water level is updated, the velocities are calculated using Equations 5.13 and 5.14. Next, as with the external mode, after a given time step has been completed, the boundary water levels and velocities of each processor are passed to its neighbor processors (Figure 5.2). Then, the parallel ELM is used to calculate the non-linear and diffusion terms at the new time level. Finally, the time step is incremented and process of solving the equations begins again.
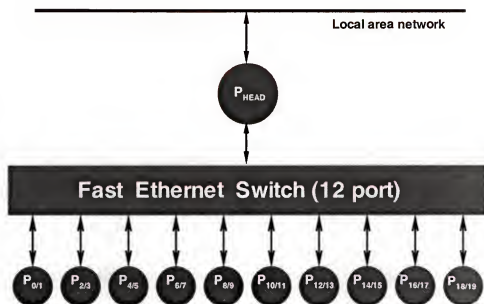
Figure 5.3: A schematic representation of the Beowulf Cluster, "nereus", built for execution of the PEM.

<div align="center">5.9   Computing Platform</div>

While the PEM will execute on any system or any collection of systems running MPI, a Beowulf Cluster (Figure 5.3) was designed and built to assess the feasibility of the Beowulf concept on estuarine and coastal environmental modeling. Beowulf Clusters are composed of Commercial Off The Shelf (COTS) PCs running the Linux operating system connected via a private network and many researchers have had success with the Beowulf concept (Sterling et al., 1995; Ridge et al., 1997).

The private network is created by placing the processing nodes behind a head node; thus, only inter-nod processing-related communication consumes network bandwidth on a dedicated ethernet switch. A description of the hardware and software used in the cluster is given in Table 5.2.

The main advantage of the Beowulf concept is the relatively low cost per processing node. One additional processor for the SGI Origin 2000 costs about \$10,000 presently, while a COTS 450 MHz Pentium III computer costs about \$1,000, a 90%

Table 5.2: Description of the hardware and software used to create the cluster.

Hardware	
Head Node	Dell PowerEdge 1300
	Dual processor Pentium III 450 Mhz
	256 MB memory, 9 GB SCSI disk drive
	Cdrom, 3.5" floppy drive
	2 10/100 megabit ethernet cards
Processing	10 Dell PowerEdge 1300
Nodes	Dual processor Pentium III 450 Mhz
	256 MB memory, 9 GB SCSI disk drive
	Cdrom, 3.5" floppy drive
	1 10/100 megabit ethernet card
Switch	3Com SuperStack II
	12 port 10/100 megabit switch

Software	
Operating System	Head Node: Red Hat Linux 6.2 (Zoot)
	Processing Nodes: Red Hat 7.1 (Seawolf)
Compilers	GNU, Portland Group 3.2-4
Message Passing	MPICH 1.2.2, PVM 3.4.2
Numerical Libraries	Aztec/Blacs/Blocksolve95/Scalapack

savings. Even though the SGI processor is faster than a Pentium processor, it is not
10x faster. Along with decreased processor cost, memory cost is also much cheaper
for a common Pentium based computer.

CHAPTER 6
VERIFICATION OF THE PEM

Before a numerical model can be applied to a real system, it is necessary to verify the accuracy of the various components and numerical schemes of the model. This chapter presents a study on the numerical accuracy of the 2D, parallel, wetting and drying numerical model, PEM, by comparing model results with a number of analytic solutions describing the circulation in idealized basins forced by wind and tide with and without Coriolis acceleration and wetting and drying. In addition, the parallel speedup of the explicit scheme and the semi-implicit and implicit schemes are presented along with a discussion of how the characteristics of the iterative solver affect the implicit calculations.

6.1 Wind Forcing

The analytical setup due to a constant wind stress in a rectangular basin can be written as

$$\zeta(x) = \frac{\tau_w}{\rho g h} \left(x - \frac{L}{2} \right) \tag{6.1}$$

where ζ is the setup of the water surface, τ_w is the applied wind stress, $H\ (= h + \zeta)$ is the total depth of the water column, L is the length of the basin and x is the distance from the left edge. The depth is chosen such that $h >> \zeta$ so that H can be approximated by h.

The computational grid used in the wind stress test is a 21x5 cell, orthogonal grid with a length, L, of 21 km and a width, W, of 5 km (Figure 6.1). The depth is a constant 5 m and the grid spacings, Δx and Δy are fixed at 1 km. A constant wind stress of 1 dyne/cm^2 is applied in the positive x-direction and a 900 s time step
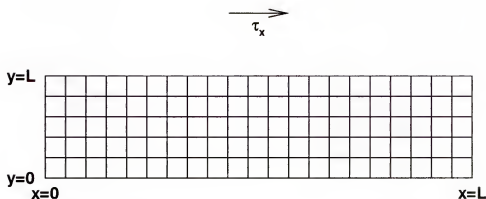
Figure 6.1: Computation grid for the simple wind forcing analytic test.

is used in the model. Table 6.1 shows the a comparison between the simulated water level and the water level determined from the analytic solution. The model simulates this analytic test case exactly.

Table 6.1: A comparison between the analytic and simulated steady state setup of water level.

x (km)	$\zeta_{theoretical}(cm)$	$\zeta_{simulated}$ (cm)
0.5	-2.04	-2.04
10.5	0	0.00
20.5	+2.04	+2.04

6.2 Tidal Forcing With Coriolis

The linearized 2D shallow water equations of continuity, x- and y-momentum without friction or diffusion take the following form

$$\frac{\partial \eta}{\partial t} + \frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} = 0 \qquad (6.2)$$

$$\frac{\partial U}{\partial t} + c^2 \frac{\partial \eta}{\partial x} - \Omega V = 0 \qquad (6.3)$$

$$\frac{\partial V}{\partial t} + c^2 \frac{\partial \eta}{\partial y} + \Omega U = 0 \qquad (6.4)$$

where where η is the surface elevation, U and V are the vertically integrated velocities in the x- and y-directions, respectively, t is time, Ω is the Coriolis parameter and

$c = \sqrt{gH}$ is the wave celerity and assuming that the depth, H is much larger than the surface elevation.

Assuming only one tidal constituent with a period $T = \frac{2\pi}{\sigma}$, these equations can be solved for U, V, and η (Rahman, 1983) and are given as

$$U(x,y,t) = \frac{ic^2\sigma\pi}{(\sigma^2-\Omega^2)x_1} \sum_{m=1}^{\infty} mH_m \left[1 + \frac{\Omega^2 k_{2m} x_1^2}{\sigma^2 m^2 \pi^2}\right] \tag{6.5}$$
$$\times \sin\left(\frac{m\pi x}{x_1}\right) \exp\left(-ik_{2m}(y_1 - y) - i\sigma t\right)$$

$$V(x,y,t) = \frac{H_o m_2 c^2}{\sigma} \left[R \exp\left(\frac{\Omega m_2 x}{\sigma} - im_2(y_1 - y)\right)\right] \exp(-i\sigma t)$$
$$- \frac{H_o m_2 c^2}{\sigma} \left[\exp\left(\frac{\Omega m_2(x_1 - x)}{\sigma} + im_2(y_1 - y)\right)\right] \exp(-i\sigma t) \tag{6.6}$$
$$+ \sum_{m=1}^{\infty} H_m \left[\frac{k_{2m}}{\sigma} \cos\left(\frac{m\pi x}{x_1}\right) + \frac{\Omega x_1}{m\pi c^2} \sin\left(\frac{m\pi x}{x_1}\right)\right]$$
$$\times \exp\left(-ik_{2m}(y_1 - y) - i\sigma t\right)$$

$$\eta(x,y,t) = H_o \exp\left(\frac{\Omega m_2(x_1 - x)}{\sigma} + im_2(y_1 - y) - i\sigma t\right)$$
$$+ RH_o \exp\left(\frac{\Omega m_2 x}{\sigma} - im_2(y_1 - y) - i\sigma t\right) \tag{6.7}$$
$$+ \sum_{m=1}^{\infty} H_m \left[\cos\left(\frac{m\pi x}{x_1}\right) + \frac{\Omega k_{2m} x_1}{\sigma m\pi} \sin\left(\frac{m\pi x}{x_1}\right)\right]$$
$$\times \exp\left(-ik_{2m}(y_1 - y) - i\sigma t\right)$$

where x_1 is the length of the basin in the x-direction, y_1 is the length of the basin in the y-direction and H_o is the amplitude of the tidal forcing at the open boundary. The wave numbers of the mth Poincare mode and that of the Kelvin are given by

$$k_{2m}^2 = -\left[\frac{m^2\pi^2}{x_1^2} - \frac{\sigma^2 - \Omega^2}{c^2}\right] \tag{6.8}$$

$$m_2^2 = \frac{\sigma^2}{c^2} \tag{6.9}$$

The unknown constants, R, H_1, H_2, ..., H_N are obtained when the equation for $V(x,y,t) = 0$ is satisfied yielding

$$H_o m_2 \left[R \exp\left(\frac{\Omega m_2 x}{\sigma}\right) - \exp\left(\frac{\Omega m_2(x_1 - x)}{\sigma}\right)\right]$$
$$+ \sum_{m=1}^{\infty} H_m \left[k_{2m} \cos\left(\frac{m\pi x}{x_1}\right) + \frac{\Omega x_1 \sigma}{m\pi c^2} \sin\left(\frac{m\pi x}{x_1}\right)\right] = 0 \tag{6.10}$$

To solve for the unknown constants, the equation is truncated at the Nth term leaving $N + 1$ unknowns. The equation is then applied to $N + 1$ points on $(0, x_1)$ resulting in $N + 1$ simultaneous equations for the $N + 1$ unknowns which are then solved. As N is increased, a converging sequence of values is found for R and each of the H_m's. The computational grid used in the wind stress test is a 40x40

cell, orthogonal grid with a length and width of 41 km (Figure 6.2). The depth is a constant 10 m and the grid spacings, Δx and Δy are fixed at 1 km. Using the analytic solution, surface elevation and velocity are imposed at the southern boundary with $\Omega = 0.00001$, $T = 12$ hours, and $H_o = 50$ cm. As can be seen in a comparison between the simulated and measured water level and velocity (Figure 6.3), the model is able to simulate this analytic test well.
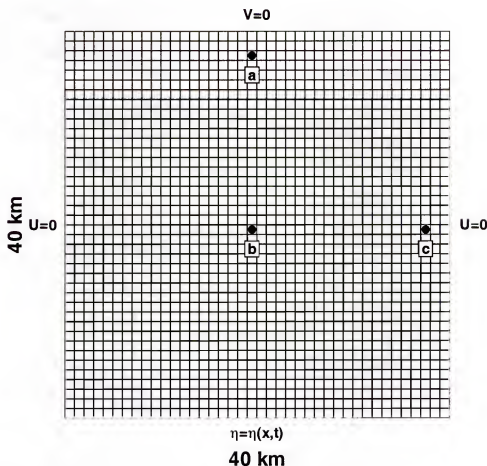
Figure 6.2: Grid system for Coriolis test. The three stations used for comparison are given the letters: a, b, and c.

6.3 Tidal Forcing in a Basin with Linearly Varying Depth

To validate the wetting and drying scheme developed, a robust analytical test needs to be performed. Carrier and Greenspan (1958) obtained the theoretical
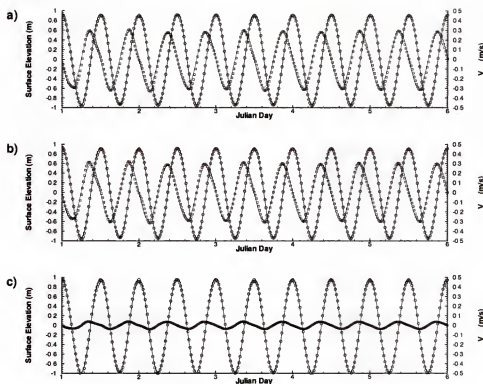
Figure 6.3: A comparison between analytic and simulated surface elevation and velocity at the three locations, a, b, and c. Solid lines indicate analytic solutions for surface elevation while dotted lines indicate analytic solutions for velocity in the y-direction. Circles indicate the simulated solution for surface elevation and squares indicate the simulated solution for velocity in the y-direction.

solution to wave propagation on a linearly sloping beach(Figure 6.4). Their solution was also used by Liu (1988), Luo (1993), Davis (1996) and Lee (2000).

The one-dimensional nonlinear shallow water equations can be written as

$$\frac{\partial \eta^*}{\partial t^*} + \frac{\partial}{\partial x^*}\left[(\eta^* + h^*)u^*\right] = 0 \tag{6.11}$$

$$\frac{\partial u^*}{\partial t^*} + u^*\frac{\partial u^*}{\partial x^*} + g\frac{\partial \eta^*}{\partial x^*} = 0 \tag{6.12}$$

where asterisks denote dimensional quantities, η is the water surface elevation above the mean water level, h is the still water depth which varies linearly with x, and u is the velocity in the x-direction. Letting L be the characteristic length scale of the
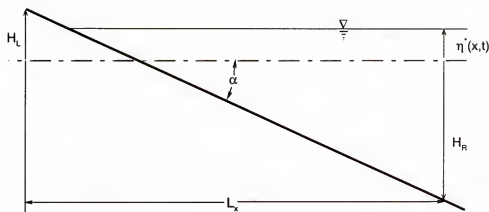
Figure 6.4: Wave propagating on a linearly sloping beach.

wave, the time and velocity scales can be defined as

$$T = \sqrt{\frac{L}{\phi g}} \tag{6.13}$$

$$u_o = \sqrt{\phi g L} \tag{6.14}$$

where ϕ is the beach angle. The equations are then non-dimensionalized using the following relations:

$$
\begin{aligned}
x &= \frac{x^*}{L} \\
t &= \frac{t^*}{T} \\
\eta &= \frac{\eta^*}{\phi L} \\
h &= \frac{h^*}{\phi L} = x \\
u &= \frac{u^*}{u_o}
\end{aligned} \tag{6.15}
$$

Defining

$$c^2 = \frac{h^* + \eta^*}{\phi L} = h + \eta = x + \eta \tag{6.16}$$

Equations 6.11 and 6.12 then become

$$\eta_t + [(\eta + x)\, u]_x = 0 \tag{6.17}$$

$$u_t + uu_x + \eta_x = 0 \tag{6.18}$$

Rewriting Equations 6.17 and 6.18 in terms of u and c gives

$$2c_t + 2uc_x + cu_x = 0 \tag{6.19}$$

$$u_t + uu_x + 2cc_x = 1 \tag{6.20}$$

Carrier and Greenspan (1958) transformed Equations 6.19 and 6.20 into a problem with only one linear equation through a series of elegant transformations. A brief derivation will be presented here. Adding and subtracting Equations 6.19 and 6.20 gives

$$\frac{d}{dt}(u2c - t) = 0 \qquad \text{along} \qquad \frac{dx}{dt} = uc \tag{6.21}$$

Defining the characteristic variables ζ and ξ as

$$\zeta = u + 2c - t \tag{6.22}$$

$$-\xi = u - 2c - t \tag{6.23}$$

Equation 6.21 becomes

$$\zeta = constant \qquad \text{along} \qquad \frac{dx}{dt} = u + c \tag{6.24}$$

$$\xi = constant \qquad \text{along} \qquad \frac{dx}{dt} = u - c \tag{6.25}$$

Assuming x and t are functions of ζ and ξ, then for $\zeta = constant$ or $\xi = constant$

$$\frac{dx}{dt} = \frac{\frac{\partial x}{\partial \xi}}{\frac{\partial t}{\partial \xi}} \qquad \text{if} \qquad \zeta = constant \tag{6.26}$$

$$\frac{dx}{dt} = \frac{\frac{\partial x}{\partial \zeta}}{\frac{\partial t}{\partial \zeta}} \qquad \text{if} \qquad \xi = constant \tag{6.27}$$

From these two equations, it can be shown that

$$x_\xi = t_\xi(u + c) \tag{6.28}$$

$$x_\zeta = t_\zeta(u + c) \tag{6.29}$$

From Equations 6.22 and 6.23, the following can be attained

$$u + c = \frac{(3\zeta - \xi)}{4} + t \tag{6.30}$$

$$u - c = \frac{(\zeta - 3\xi)}{4} + t \tag{6.31}$$

Substituting Equations 6.30 and 6.31 into Equations 6.28 and 6.29 yields the following transform relationships between (x, t) and (ζ, ξ):

$$x_\xi = \frac{t_\xi(3\zeta - \xi)}{4} + \left(\frac{t^2}{2}\right)_\xi \tag{6.32}$$

$$x_\zeta = \frac{t_\zeta(\zeta - 3\xi)}{4} + \left(\frac{t^2}{2}\right)_\zeta \tag{6.33}$$

Eliminating x from Equations 6.32 and 6.33 results in

$$2(\zeta + \xi)t_{\zeta\xi} + 3(t_\zeta + t_\xi) = 0 \tag{6.34}$$

a linear partial differential equation. It is convenient to introduce new variables σ and λ defined as

$$\lambda = \xi - \zeta = 2(t - u) \tag{6.35}$$

$$\sigma = \xi + \zeta = 4c \tag{6.36}$$

Equation 6.34 then becomes

$$t_{\lambda\lambda} = t_{\sigma\sigma} + \frac{3t_\sigma}{\sigma} \tag{6.37}$$

Since $t = \frac{\lambda}{2} + u$ from Equation 6.35, u must also satisfy Equation 6.37, thus

$$u_{\lambda\lambda} = u_{\sigma\sigma} + \frac{3u_\sigma}{\sigma} \tag{6.38}$$

Introducing a "potential", $\varphi(\sigma, \lambda)$, defined as

$$u = \frac{\varphi_\sigma}{\sigma} \tag{6.39}$$

Equation 6.38 becomes

$$\varphi_{\lambda\lambda} = \varphi_{\sigma\sigma} + \frac{\varphi_\sigma}{\sigma} \tag{6.40}$$

Equation 6.40 is a single partial differential equation whose boundary condition at the shoreline is $\sigma = 0$, which corresponds to the condition $c = 0$, *i.e.*, the total water depth at the shoreline must equal zero at all times.

Using the variables σ, λ, and the potential, $\varphi(\sigma, \lambda)$, Carrier and Greenspan (1958) proposed the following expressions for t, x, η, u, and c:

$$t = \frac{\lambda}{2} + u = \frac{\lambda}{2} + \frac{\varphi_\sigma}{\sigma} \tag{6.41}$$

$$x = \frac{u^2}{2} + c^2 + \frac{\varphi_\lambda}{4} = \frac{1}{2}\left(\frac{\varphi_\sigma}{\sigma}\right)^2 + \frac{\varphi_\lambda}{4} + \frac{\sigma^2}{16} \tag{6.42}$$

$$\eta = c^2 - x = \frac{\sigma^2}{16} - x = -\frac{\varphi_\lambda}{4} - \frac{\sigma^2}{16} \tag{6.43}$$

$$u = \frac{\varphi_\sigma}{\sigma} \tag{6.44}$$

$$c = \frac{\sigma}{4} \tag{6.45}$$

If $\varphi(\sigma, \lambda)$ is given, then Equations 6.41 through 6.45 give t, x, η, u, and c parametrically in terms of σ and λ. In general, it is difficult to obtain direct functional relationships for η and u in terms of x and t.

Carrier and Greenspan (1958) point out a solution to Equation 6.40 is

$$\varphi(\sigma, \lambda) = -8A_o J_0\left(\frac{\sigma}{2}\right)\sin\left(\frac{\lambda}{2}\right) \tag{6.46}$$

where A_o is an arbitrary amplitude parameter and J_0 is a zeroth order Bessel function of the first kind. This potential represents a standing wave solution resulting from the perfect reflection of a unit frequency wave. With $\varphi(\sigma, \lambda)$ given, Equations 6.41 through 6.45 will implicitly give the solution of the standing wave.

To evaluate $\eta(x, t)$ and $u(x, t)$ for a given x and t, Equations 6.41 through 6.45 must be solved numerically. For specific values of x and t, σ and λ are determined using a least squares method so that $\eta(x, t)$ and $u(x, t)$ are easily obtained from Equations 6.43 and 6.44, respectively.

The length of the basin, L_x, is 62 km and the width 10 km. The bottom slope, α, is 1:2500. The height above still water on the left side, H_L is 2 m and the depth

below still water on the right side, H_R is 24 m. The period of the long wave is 1 hr or a frequency, ω^*, of 0.001745 1/s. The characteristic length scale, L is defined by

$$L = \frac{\phi g}{(\omega^*)^2} = 1288 \text{ m} \tag{6.47}$$

which yields the velocity scale

$$u_o = \sqrt{\phi g L} = 2.25 \text{ m/s} \tag{6.48}$$

For the numerical simulation, a 620×5 rectangular grid ($\Delta x = 100$ m, $\Delta y = 2000$ m) is used along with a time step of $\Delta t = 30$ s is chosen. The degree of implicitness, θ_1, is chosen to be 0.55 and the tolerance of the conjugate gradient solver is 10^{-6}. At $t = 0$ an initial wave form, as calculated by the theoretical solution, is applied to the surface elevation. For $t > 0$, the wave amplitude at the offshore boundary, $\eta^*(t)$ is given by

$$\eta^*(t) = \eta(t)\phi L \tag{6.49}$$

where $\eta(t)$ is the dimensionless value of the wave amplitude obtained from the theoretical solution. After three periods, the numerical solution of the standing wave is obtained and compared to the theoretical solution.

Figures 6.5 and 6.6 show comparisons between simulated and theoretical wave profiles over the length of the basin during 7 time instants of a half period. The model is able to simulate the wetting and drying of the shoreline well.

6.4 Conjugate Gradient Tolerance Analysis

As was discussed in Chapter 5, an iterative solver is used to solve the 9-diagonal matrix (Equation 5.16) for surface elevation which results from a semi-implicit or implicit discretization. This matrix is solved using Aztec (Tuminaro et al., 1999), a parallel iterative solver. The iterative solving algorithm ceases if a residual (Equation 5.43) is less than a specified tolerance. To determine the effect of changing this tolerance on the outcome of the numerical solver, several simulations were
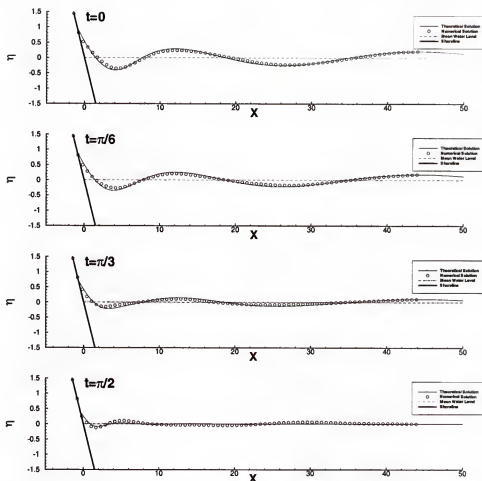
Figure 6.5: Non-dimensional wave profiles as predicted by theory and the numerical model for times $t = 0$ through $t = \pi/2$.

performed using different tolerances. The simulation chosen was the analytic test for tidal forcing with Coriolis discussed previously. Comparisons between the simulated water level and velocities using the very small default tolerance (10^{-6}) and larger ones are shown in Table 6.2. The results show that a tolerance as large 10^{-4} changes the results by less than $10^{-4}\%$.
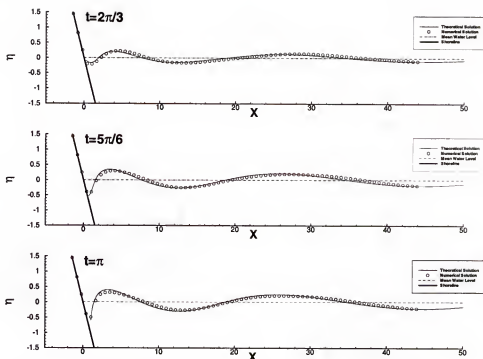
Figure 6.6: Non-dimensional wave profiles as predicted by theory and the numerical model for times $t = 2\pi/3$ through $t = \pi$.

Table 6.2: The RMS difference ($/10^{-4}$) between the surface elevation, η, u-averaged, U, and v-averaged, V, velocities calculated with a conjugate gradient tolerance of 10^{-6} and the results calculated with tolerances of 10^{-2}, 10^{-3}, 10^{-4}, and 10^{-5}. Results shown are for the 5 day simulation used to compare with the analytic test for Coriolis and are given in units of m and m/s, respectively.

Station	Variable	Tolerance				
		10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
a	η	5.41	1.35	0.57	0.17	0.00
b	U	1.49	0.29	0.12	0.00	0.00
c	V	3.07	0.71	0.24	0.17	0.00
a	η	4.52	1.25	0.47	0.17	0.00
b	U	1.20	0.37	0.29	0.24	0.00
c	V	5.00	1.34	0.49	0.37	0.00
a	η	4.57	1.28	0.49	0.12	0.00
b	U	2.55	0.59	0.12	0.24	0.00
c	V	5.08	1.45	0.60	0.51	0.00

6.5   Parallel Timing Analysis

To determine how well the model performs in parallel, simulations are performed using various processor and simulation configurations on the explicit, semi-implicit and implicit model formulations. In addition, the conjugate gradient algorithm in Aztec (Tuminaro et al., 1999) can be modified; thus, affecting the simulation time. This simulation chosen as a basis for experimentation was the simple wind setup test described previously. Figure 6.7 shows the grid configuration for a multi-processor case with the x-direction split among four processors.
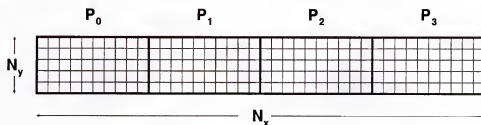
Figure 6.7: Four processor grid configuration for the simulations used to calculate speedup with the PEM. The number of cells in the x- and y-directions are N_x and N_y, respectively.

6.5.1   Explicit Mode

In general, explicit discretization of an equations yields equations which are easily solved in parallel. Difficulty and loss of speed occurs only at the end of a given time step when information must be passed to other processors. However, due to stringent time step limitations ($\Delta t \leq \Delta x/u$), explicit formulations are difficult to use in practical simulations which requite fast simulation times and small grid spacings. Table 6.3 illustrates the CPU times for various grid sizes using a 1 s time step which are then plotted in Figure 6.8. These results show how the speedup improves with a larger number of grid cells in the x-direction; however, as the number of cells in the y-direction gets larger (the direction through which data is passed to other processors), the speedup decreases.

Table 6.3: CPU time per iteration of the parallel, message passing explicit PEM using various grid sizes. Parallel speedup is shown in parenthesis.

N_z	N_y	# of Iterations	$n = 1$	$n = 2$	$n = 4$	$n = 5$	$n = 8$	$n = 10$	$n = 20$
200	25	3600	0.01(1.00)	0.01(1.89)	0.01(2.25)	0.01(2.25)	0.01(1.49)	0.01(1.27)	0.03(0.38)
400	25	3600	0.03(1.00)	0.01(2.08)	0.02(3.23)	0.01(3.11)	0.01(3.01)	0.01(2.62)	0.03(0.84)
800	25	3600	0.06(1.00)	0.03(1.98)	0.02(3.81)	0.01(4.46)	0.01(5.18)	0.01(4.84)	0.03(1.77)
1600	25	3600	0.14(1.00)	0.06(2.22)	0.03(4.45)	0.03(5.38)	0.02(7.48)	0.02(8.35)	0.04(3.71)
3200	25	900	0.28(1.00)	0.14(1.95)	0.06(4.41)	0.05(5.22)	0.03(8.18)	0.03(9.55)	0.04(6.44)
6400	25	900	0.57(1.00)	0.29(1.99)	0.15(3.95)	0.11(5.21)	0.07(8.73)	0.06(10.09)	0.06(10.40)
12800	25	900	1.14(1.00)	0.60(1.92)	0.29(3.92)	0.23(4.87)	0.15(7.70)	0.11(10.04)	0.08(14.15)
25600	25	900	2.32(1.00)	1.19(1.95)	0.60(3.87)	0.47(4.89)	0.29(7.88)	0.24(9.78)	0.14(16.98)
51200	25	900	4.67(1.00)	2.40(1.95)	1.19(3.93)	0.98(4.78)	0.60(7.78)	0.48(9.79)	0.25(18.63)
200	25	3600	0.01(1.00)	0.01(1.97)	0.01(2.34)	0.01(2.37)	0.01(1.57)	0.01(1.37)	0.03(0.40)
200	50	3600	0.03(1.00)	0.01(2.05)	0.01(3.17)	0.01(3.40)	0.01(2.91)	0.01(2.65)	0.03(0.83)
200	100	3600	0.06(1.00)	0.03(1.99)	0.02(3.52)	0.01(4.06)	0.01(4.77)	0.01(4.75)	0.03(1.71)
200	200	3600	0.14(1.00)	0.06(2.30)	0.03(4.26)	0.03(5.18)	0.02(6.86)	0.02(7.61)	0.02(6.02)
200	400	900	0.28(1.00)	0.15(1.91)	0.07(4.10)	0.06(4.94)	0.04(6.97)	0.04(7.90)	0.05(5.80)
200	800	900	0.55(1.00)	0.29(1.90)	0.15(3.58)	0.13(4.31)	0.08(7.16)	0.07(8.19)	0.05(11.37)
200	1600	900	1.11(1.00)	0.58(1.90)	0.31(3.56)	0.26(4.28)	0.17(6.35)	0.15(7.53)	0.09(12.05)
200	3200	900	2.23(1.00)	1.17(1.90)	0.62(3.57)	0.52(4.31)	0.35(6.40)	0.29(7.63)	0.19(11.90)
200	6400	900	4.44(1.00)	2.34(1.90)	1.93(2.30)	1.56(2.85)	1.04(4.27)	0.86(5.16)	0.50(8.80)
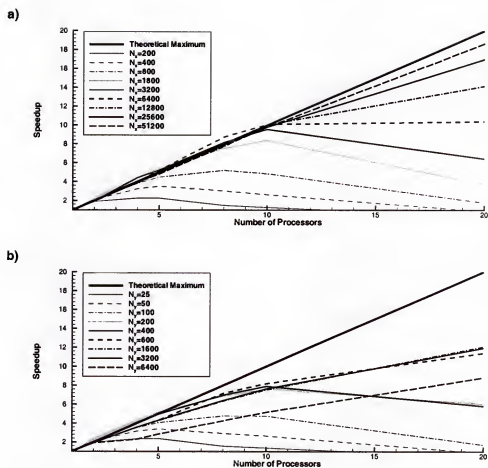
Figure 6.8: Parallel speedup using the explicit PEM and various grid configurations. The time steps are 1 s and the number of cells in the opposite direction is fixed at a) $N_y = 25$ and b) $N_x = 200$.
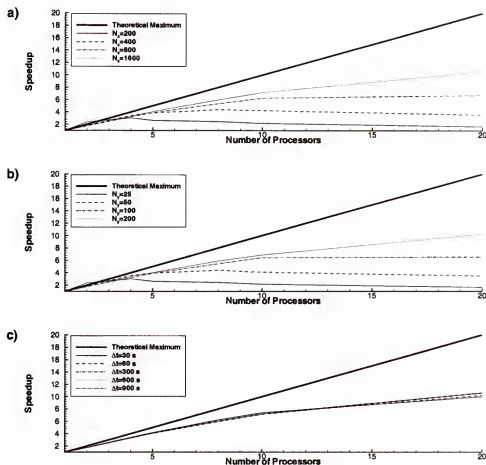
Figure 6.9: Parallel speedup using the semi-implicit PEM using various grid sizes and time steps. The size of the grid and time step are a) $N_y = 25$ and $\Delta t = 900\ s$, b) $N_x = 25$ and $\Delta t = 900\ s$ and c) $N_x = 1600$ and $N_y = 25$.

6.5.2 Semi-Implicit and Implicit Modes

While the semi-implicit and implicit discretizations is more difficult to parallelize, they allow much larger time steps to simulate the same explicit scenarios. Table 6.4 and Figure 6.9 show the speedup using different grid configurations and time steps.

In addition to varying simulation parameters, parallel timing simulations were performed using the various configurations available with Aztec's iterative solver. Results of these simulations are shown in Table 6.5 and Figure 6.10.

Table 6.4: CPU time per iteration of the parallel, message passing semi-implicit PEM for various grid sizes and time steps. Parallel speedup is shown in parenthesis.

N_z	N_y	Δt (s)	$n = 1$	$n = 2$	$n = 4$	$n = 5$	$n = 8$	$n = 10$	$n = 20$
200	25	900	2.95(1.00)	1.24(2.37)	0.97(3.03)	1.12(2.63)	1.20(2.45)	1.37(2.16)	1.79(1.65)
400	25	900	12.02(1.00)	6.66(1.80)	3.50(3.43)	3.14(3.83)	2.74(4.39)	2.84(4.24)	3.39(3.54)
800	25	900	33.66(1.00)	19.41(1.73)	10.49(3.21)	8.66(3.89)	6.28(5.36)	5.40(6.24)	5.01(6.72)
1600	25	900	71.65(1.00)	41.64(1.72)	21.71(3.30)	17.68(4.05)	12.04(5.95)	10.05(7.13)	6.78(10.57)
200	25	900	2.95(1.00)	1.24(2.37)	0.97(3.03)	1.12(2.63)	1.20(2.45)	1.37(2.16)	1.79(1.65)
200	50	900	7.21(1.00)	3.90(1.85)	2.02(3.58)	1.83(3.94)	1.64(4.41)	1.76(4.09)	2.07(3.48)
200	100	900	16.04(1.00)	8.96(1.79)	4.96(3.23)	4.11(3.90)	2.94(5.45)	2.49(6.45)	2.45(6.54)
200	200	900	37.46(1.00)	21.64(1.73)	11.01(3.40)	9.34(4.01)	6.37(5.88)	5.47(6.85)	3.67(10.20)
1600	25	30	2.91(1.00)	1.65(1.76)	0.86(3.39)	0.70(4.16)	0.47(6.25)	0.39(7.40)	0.29(9.99)
1600	25	60	4.94(1.00)	2.84(1.74)	1.47(3.37)	1.20(4.12)	0.81(6.11)	0.68(7.26)	0.48(10.21)
1600	25	300	22.37(1.00)	13.01(1.72)	6.67(3.35)	5.46(4.09)	3.66(6.11)	3.08(7.26)	2.11(10.62)
1600	25	600	45.64(1.00)	26.63(1.71)	13.89(3.28)	11.21(4.07)	7.59(6.01)	6.35(7.19)	4.30(10.62)
1600	25	900	71.65(1.00)	41.64(1.72)	21.71(3.30)	17.68(4.05)	12.04(5.95)	10.05(7.13)	6.78(10.57)

Table 6.5: CPU time per iteration of the parallel, semi-implicit message passing PEM using various iterative solver configurations available with the Aztec parallel iterative library (Tuminaro et al., 1999). Parallel speedup is shown in parenthesis. The solving algorithms are conjugate gradient squared (CGS), transpose-free quasi-minimal residual (TFQMR) and bi-conjugate gradient with stabilization (BICGSTAB). The scaling options are point Jacobi scaling (Point Jacobi), scale each row so the magnitude of its elements sum to 1 (Row Sum), symmetric scaling so diagonal elements are 1 (Sym. Diagonal) and symmetric scaling using the matrix row sums (Sym. Row Sum).

Algorithm	Scaling	Tolerance	$n = 1$	$n = 2$	$n = 4$	$n = 5$	$n = 8$	$n = 10$	$n = 20$
CGS	Row Sum	10^{-6}	71.65(1.00)	41.64(1.72)	21.71(3.30)	17.68(4.05)	12.04(5.95)	10.05(7.13)	6.78(10.57)
TFQMR	Row Sum	10^{-6}	80.97(1.00)	47.38(1.71)	24.50(3.31)	20.21(4.01)	14.06(5.76)	12.17(6.65)	8.92(9.08)
BICGSTAB	Row Sum	10^{-6}	59.93(1.00)	37.91(1.58)	20.10(2.98)	16.93(3.54)	11.31(5.30)	9.83(6.10)	7.59(7.90)
CGS	Point Jacobi	10^{-6}	72.59(1.00)	42.38(1.71)	21.96(3.31)	18.07(4.02)	12.00(6.05)	10.17(7.14)	6.76(10.73)
CGS	Row Sum	10^{-6}	71.65(1.00)	41.64(1.72)	21.71(3.30)	17.68(4.05)	12.04(5.95)	10.05(7.13)	6.78(10.57)
CGS	Sym. Diagonal	10^{-6}	73.60(1.00)	42.84(1.72)	22.33(3.30)	18.22(4.04)	12.21(6.03)	10.23(7.20)	7.04(10.46)
CGS	Sym. Row Sum	10^{-6}	73.46(1.00)	42.91(1.71)	22.43(3.27)	18.26(4.02)	12.19(6.03)	10.41(7.06)	6.81(10.78)
CGS	Row Sum	10^{-2}	31.33(1.00)	18.20(1.72)	9.32(3.36)	7.57(4.14)	5.21(6.02)	4.35(7.20)	3.01(10.41)
CGS	Row Sum	10^{-3}	43.46(1.00)	25.62(1.70)	12.97(3.35)	10.62(4.09)	7.36(5.90)	6.07(7.16)	4.09(10.62)
CGS	Row Sum	10^{-4}	55.76(1.00)	32.52(1.71)	16.95(3.29)	13.68(4.08)	9.41(5.92)	7.96(7.00)	5.22(10.69)
CGS	Row Sum	10^{-5}	64.13(1.00)	37.11(1.73)	19.44(3.30)	15.96(4.02)	10.67(6.01)	8.97(7.15)	6.26(10.25)
CGS	Row Sum	10^{-6}	71.65(1.00)	41.64(1.72)	21.71(3.30)	17.68(4.05)	12.04(5.95)	10.05(7.13)	6.78(10.57)
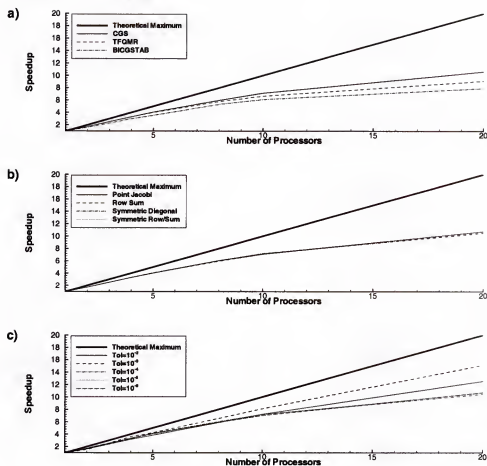
Figure 6.10: Parallel speedup using the semi-implicit PEM and various iterative solve configurations. The algorithm, scaling and tolerance are a) Row Sum and 10^{-6}, b) CGS and 10^{-6} and c) CGS and Row Sum.
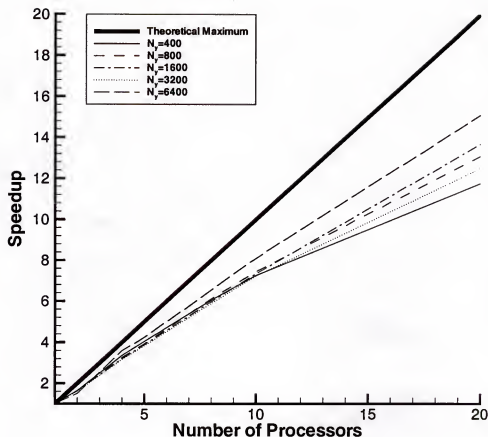
Figure 6.11: Parallel speedup using the semi-implicit PEM for large values of N_y where $N_x = 200$ and $\Delta t = 900$. Results are calculated for one time step

As can be seen from the previous results, efficiencies increase with larger grid sizes. Table 6.6 and Figure 6.11 show a series of simulations performed on large grid systems illustrating this point. For the 200×6400 (1.28 million computational cells) and 20 processors, a speedup of 15.10 is reached.

Table 6.6: CPU time per iteration of the parallel, message passing semi-implicit PEM for 1 time step and large values of N_y where $N_z = 200$ and $\Delta t = 900$. Parallel speedup is shown in parenthesis.

N_y	$n = 1$	$n = 2$	$n = 4$	$n = 5$	$n = 8$	$n = 10$	$n = 20$
400	62.33(1.00)	37.80(1.65)	18.53(3.36)	15.43(4.04)	10.31(6.04)	8.59(7.25)	5.29(11.77)
800	124.41(1.00)	76.13(1.63)	38.14(3.26)	31.75(3.92)	20.36(6.11)	16.72(7.44)	9.50(13.10)
1600	241.01(1.00)	147.55(1.63)	75.20(3.21)	62.03(3.89)	40.40(5.96)	32.84(7.34)	17.60(13.70)
3200	469.73(1.00)	291.78(1.61)	148.36(3.17)	122.74(3.83)	79.90(5.88)	65.02(7.22)	37.56(12.51)
6400	1016.77(1.00)	675.85(1.50)	282.78(3.60)	239.88(4.24)	155.46(6.54)	125.27(8.12)	67.33(15.10)

CHAPTER 7
APPLICATION OF THE PEM

In this chapter, applications of the coupled hydrodynamic and storm models of the newly developed PEM are presented. Using measured water level data in the IRL, simulations were performed to verify the model using two hurricanes formed during the 1999 Atlantic Hurricane season. Additionally, simulations of hypothetical hurricanes were performed to study the flooding caused by a hurricane making a direct landfall on the lagoon.

7.1 Historical Hurricane Information

To effectively simulate the water level due to a large storm like a hurricane, it is necessary to have a good description of the storm itself. The primary source of hurricane data comes from the National Hurricane Center Branch of NOAA's Tropical Prediction Center[1].

Based on the NOAA data in Jarrell et al. (1992), Landsea (2000) created an online version of the 20th century hurricane record for each U.S. coastal state and county. Using this data, Figure 7.1 shows all of the hurricanes which made landfall in Volusia[2], Brevard[3] and Indian River[4] Counties during the twentieth century. Since 1900, no hurricanes larger than Category 2 have made landfall in these three counties containing the IRL.

---

[1] http://www.nhc.noaa.gov/

[2] http://www.aoml.noaa.gov/hrd/tcfaq/counties/FL_Volusia.gif

[3] http://www.aoml.noaa.gov/hrd/tcfaq/counties/FL_Brevard.gif

[4] http://www.aoml.noaa.gov/hrd/tcfaq/counties/FL_IndianRiver.gif
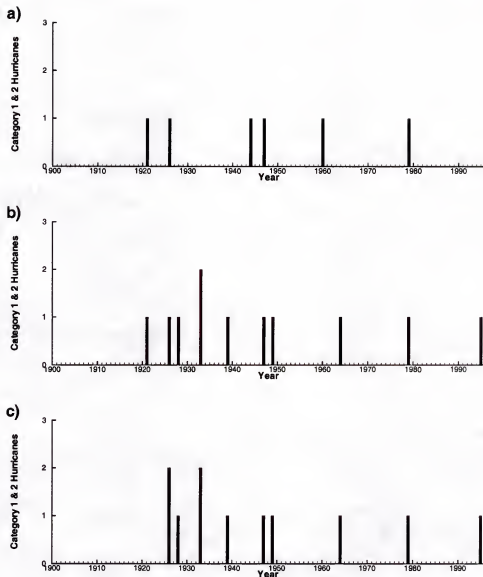
Figure 7.1: Number of hurricanes which made landfall in a) Volusia County, b) Brevard County and c) Indian River County per year. All hurricanes illustrated are minor (Category 1 and 2) (Landsea, 2000).

Hurricane tracks can be obtained from the "Atlantic Tracks File" which contains 6-hourly (00:00, 06:00, 12:00, and 18:00 UTC) center locations, maximum surface wind speeds and minimum central pressure for Atlantic Ocean tropical cyclones from 1851 to the present. More information on this dataset can be found in Jarvinen et al. (1984).

Another source of useful information are the National Hurricane Center's Preliminary Reports. These reports summarize the life history and effects of Atlantic and eastern Pacific tropical cyclones. It contains a summary of the cyclone life cycle and pertinent meteorological data, including the post-analysis best track (six-hourly positions and intensities) and other meteorological statistics. It also contains a description of damage and casualties the system produced, as well as information on forecasts and warnings associated with the cyclone. The NHC writes a preliminary report on every tropical cyclone in its area of responsibility.

Atmospheric measurements of the hurricane itself can be obtained from the "Hurricane Hunter" aircraft flown out of MacDill Air Force Base (Aircraft Operations Center[5]). These airplanes penetrate the eyewall repeatedly at altitudes up to 20,000 ft. and as low as 1,500 feet. Collected information is sent to the National Centers for Environmental Prediction (NCEP) in Camp Springs, Maryland, to be used in numerical computer models. NCEP then sends the model data back to forecasters at the National Hurricane Center. The information is important for current hurricane predictions as well as for research purposes.

The aircraft probes every wind and pressure change again and again during the course of a 10-hour mission. The scientists aboard the aircraft deploy instruments called GPS- (Global Positioning System) dropwindsondes as the airplanes fly through the hurricane. These devices continuously radio back measurements ("Vortex Data Messages") of pressure, humidity, temperature, and wind direction and speed as the instruments fall toward the sea, providing a detailed look at the intensity and structure of the storm (Table 7.1).

7.2 1999 Atlantic Hurricane Season

Although no hurricanes made direct landfall in the IRL during the 1999 Atlantic hurricane season, the tidal record of water level at Sebastian Inlet (FDEP

---

[5]http://www.omao.noaa.gov/aoc/index.html

Table 7.1: Information contained in a Vortex Data Message.

Line	Description
(A)	Date and time of the vortex fix
(B)	Position of the vortex fix
(C)	Minimum height of a standard pressure level
(D)	Estimate of maximum surface wind
(E)	Bearing and range to the maximum surface wind from the center
(F)	Maximum flight level wind near storm center
(G)	Bearing and range from the storm center to the maximum flight level wind
(H)	Minimum sea level pressure
(I)	Maximum flight level temperature and pressure altitude outside the eye
(J)	Maximum flight level temperature and pressure altitude inside the eye
(K)	Dew point and sea surface temperature inside the eye
(L)	Eye character
(M)	Eye shape orientation and diameter
(N)	Confirmation of position and time fix
(O)	Fix determined by / fix level
(P)	Navigation fix accuracy
(Q)	Remarks section

#872-2004) (Figure 7.2) shows that two storms came close enough to cause a noticeable storm surge, once in September and once in October. These surges correspond to the passage of Hurricanes Floyd and Irene, respectively. The tracks of these storms were obtained from the Atlantic Tracks File and are shown in Figure 7.3.

7.2.1   Hurricane Floyd

The following characterization of Hurricane Floyd (September 7-17, 1999) was taken from Pasch et al. (1999).

> Floyd was a large and intense Cape Verde hurricane that pounded the central and northern Bahama islands, seriously threatened Florida, struck the coast of North Carolina and moved up the United States east coast into New England. It neared the threshold of category five intensity on the Saffir/Simpson Hurricane Scale as it approached the Bahamas, and produced a flood disaster of immense proportions in the eastern United States, particularly in North Carolina.
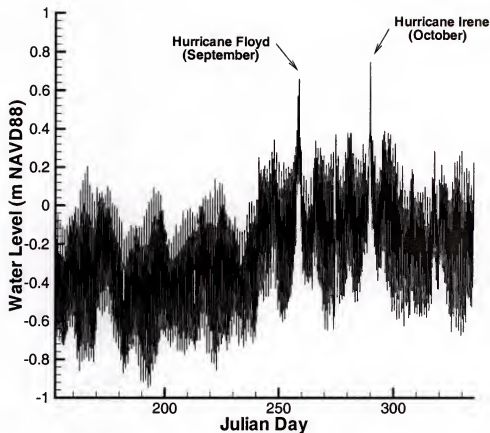
Figure 7.2: Measured water level at Sebastian Inlet (FDEP #872-2004) during the 1999 Atlantic hurricane season (June-November).

Table 7.2 shows the history of Hurricane Floyd with all of the characteristics, except for radius to maximum wind speed which was obtained from the previously mentioned Atlantic Tracks File. The radius of maximum wind speed was calculated from the Vortex Data Messages. The Vortex Data Message measurements are not taken at any specific time and represent instantaneous conditions, so the messages were converted to hourly values and then subjected to a 5-hour moving average.

7.2.2 Hurricane Irene

The following characterization of Hurricane Irene (October 13-19, 1999) was taken from Avila (1999).

> Irene was a typical wet October tropical cyclone that moved over the Florida Keys and southeast Florida dumping from 10 to 20 inches of rain.

Table 7.2: History of Hurricane Floyd (September 1999). R is the radius to the maximum wind speed, V_{max} and ΔP is the pressure drop. The Atlantic Tracks File is the source of all attributes except pressure drop which comes from Vortex Data Messages.

Date	Time	Longitude	Latitude	UTM		ΔP	R	V_{max}
(Julian Day)	(UTC)	(deg W)	(deg N)	X(m)	Y(m)	(in hg)	(n mi)	(knots)
9/7(250)	18:00	45 35 00	14 36 00	4544772	1962378	-0.154		25
9/8(251)	00:00	46 54 00	15 00 00	4368089	1984941	-0.183		30
	06:00	48 12 00	15 18 00	4198678	1995468	-0.242		35
	12:00	49 35 00	15 48 00	4017494	2030311	-0.301		40
	18:00	51 05 00	16 17 00	3825713	2061059	-0.390		45
9/9(252)	00:00	52 35 00	16 42 00	3638538	2084194	-0.390		45
	06:00	53 54 00	17 06 00	3476094	2109350	-0.301		45
	12:00	55 05 00	17 17 00	3335104	2111219	-0.301	101	50
	18:00	56 17 00	17 53 00	3186679	2163461	-0.508	81	60
9/10(253)	00:00	57 12 00	18 17 00	3075605	2196475	-0.537	60	60
	06:00	58 12 00	18 36 00	2957561	2218544	-0.685	48	60
	12:00	58 47 00	19 17 00	2881406	2290169	-0.715	49	70
	18:00	59 35 00	20 12 00	2778180	2385504	-1.128	29	70
9/11(254)	00:00	60 24 00	20 47 00	2679401	2441391	-1.246	27	80
	06:00	61 05 00	21 23 00	2595784	2501087	-1.482	21	95
	12:00	62 00 00	21 53 00	2489300	2545919	-1.512	20	95
	18:00	63 00 00	22 30 00	2373319	2603250	-1.394	33	90
9/12(255)	00:00	64 05 00	22 42 00	2255481	2612213	-1.364	39	85
	06:00	65 11 00	22 47 00	2138161	2608504	-1.571	29	95
	12:00	66 11 00	23 00 00	2030372	2621888	-1.719	25	105
	18:00	67 24 00	23 12 00	1900831	2631857	-2.162	20	115
9/13(256)	00:00	68 41 00	23 23 00	1765373	2640357	-2.427	19	125
	06:00	70 00 00	23 36 00	1627003	2653468	-2.693	15	135
	12:00	71 24 00	23 53 00	1480423	2674625	-2.723	18	135
	18:00	72 54 00	24 06 00	1325079	2689067	-2.664	14	125
9/14(257)	00:00	74 00 00	24 30 00	1210379	2727486	-2.634	17	115
	06:00	75 18 00	24 53 00	1076356	2763962	-2.545	25	105
	12:00	76 18 00	25 23 00	973140	2815566	-2.457	37	105
	18:00	77 00 00	26 06 00	900179	2892747	-2.457	33	110
9/15(258)	00:00	77 41 00	27 06 00	828886	3001683	-2.368	28	115
	06:00	78 30 00	28 12 00	745399	3121716	-2.309	31	110
	12:00	78 54 00	29 17 00	704008	3241029	-2.073	34	100
	18:00	79 05 00	30 36 00	683756	3386656	-1.955	34	95
9/16(259)	00:00	78 41 00	32 05 00	718654	3551833	-1.866	28	90
	06:00	78 00 00	33 42 00	778069	3732741	-1.689	31	90
	12:00	76 48 00	35 42 00	880083	3958613	-1.364	61	70
	18:00	75 18 00	38 00 00	1000658	4220970	-1.158		60
9/17(260)	00:00	73 30 00	40 35 00	1135012	4519416	-0.980		50
	06:00	72 05 00	42 05 00	1237812	4697458	-0.892		50
	12:00	70 35 00	43 17 00	1345423	4845056	-0.862		45
	18:00	68 54 00	44 12 00	1467035	4965603	-0.833		45
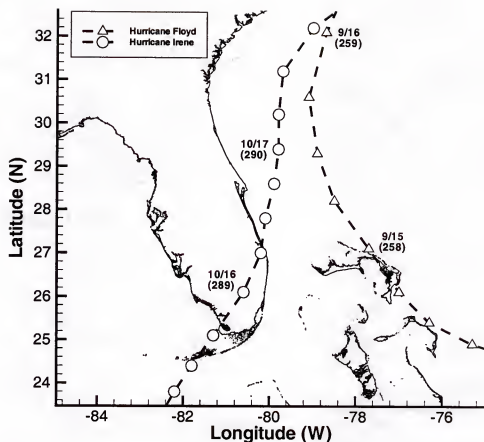
Figure 7.3: Tracks of Atlantic Hurricanes Floyd and Irene (1999).

This resulted in severe flooding conditions. This type of tropical cyclone was a common phenomena during the 30's and 40's.

Table 7.3 show the history of Hurricane Irene with the characteristics obtained in the same manner as Hurricane Floyd.

7.3   Parallel Grid Systems and Boundary Conditions

Storm surge simulations were performed using the previously developed (Chapter 5) and verified PEM (Chapter 6). Before simulations began, two parallel grid systems were created along with an appropriate set of boundary conditions to simulate Hurricanes Floyd and Irene.

Table 7.3: History of Hurricane Irene (October 1999). R is the radius to the maximum wind speed, V_{max} and ΔP is the pressure drop. The Atlantic Tracks File is the source of All attributes except pressure drop which comes from Vortex Data Messages.

Date (Julian Day)	Time (UTC)	Longitude (deg W)	Latitude (deg N)	UTM X(m)	UTM Y(m)	ΔP (in hg)	R (n mi)	V_{max} (knots)
10/12(285)	12:00	82 00 00	15 53 00	392940	1756179	-0.213		20
	18:00	83 00 00	16 23 00	286390	1812280	-0.213		20
10/13(286)	00:00	83 11 00	16 42 00	267182	1847531	-0.242		20
	06:00	83 18 00	17 17 00	255493	1912242	-0.272		30
	12:00	83 24 00	18 30 00	246592	2047067	-0.301		35
	18:00	83 35 00	19 47 00	229327	2189441	-0.360		45
10/14(287)	00:00	83 35 00	20 42 00	230911	2290956	-0.419	33	55
	06:00	83 35 00	21 00 00	231445	2324181	-0.419	26	60
	12:00	82 54 00	21 17 00	302880	2354553	-0.478	27	60
	18:00	82 24 00	22 23 00	355870	2475784	-0.537	28	60
10/15(288)	00:00	82 35 00	23 06 00	337840	2555321	-0.744	49	60
	06:00	82 11 00	23 47 00	379433	2630589	-0.744	73	65
	12:00	81 48 00	24 23 00	418871	2696747	-0.774	50	65
	18:00	81 18 00	25 06 00	469751	2775896	-0.803	44	65
10/16(289)	00:00	80 35 00	26 06 00	541665	2886662	-0.803	32	65
	06:00	80 11 00	27 00 00	581029	2986530	-0.833	33	65
	12:00	80 05 00	27 47 00	590313	3073368	-0.921	55	65
	18:00	79 54 00	28 36 00	607555	3163990	-0.862	54	65
10/17(290)	00:00	79 48 00	29 23 00	616453	3250878	-0.862	80	65
	06:00	79 48 00	30 12 00	615511	3341375	-0.833	77	65
	12:00	79 41 00	31 12 00	625442	3452329	-0.862	60	65
	18:00	79 00 00	32 12 00	688520	3564170	-1.039	42	70
10/18(291)	00:00	77 24 00	33 24 00	834866	3701236	-1.099	33	80
	06:00	75 11 00	34 47 00	1032509	3864271	-1.453		95
	12:00	71 35 00	36 47 00	1341320	4112249	-1.571		90
	18:00	67 24 00	39 00 00	1679930	4405628	-1.335		80
10/19(292)	00:00	61 00 00	41 30 00	2172925	4791818	-1.335		80
	06:00	51 30 00	44 54 00	2825303	5413844	-1.335		80
	12:00	48 00 00	48 00 00	2940097	5867213	-1.335		80
	18:00	45 00 00	51 00 00	2980858	6293166	-1.335		80

7.3.1   The IRL PEM Parallel Grid System

Two rotated coordinate systems (Figure 7.4), 20 processor PEM parallel grid system were developed to study circulation in the IRL. The first 280×128 grid system was developed using a constant Δx and Δy of 625 m and will be referred to from here on as the "coarse grid". This grid system was divided in the x-direction such that each of the 20 processors solved a 14×128 section of the overall grid. The second

1400 × 640 grid system was developed by refining the first grid system five times in both the x- and y-directions (Figure 7.5) and will be referred to from here on as the "fine grid". This grid system has a constant Δx and Δy of 125 m and was divided in the x-direction such that each of the 20 processors solved a 70 × 640 section of the overall grid. Each large box in the figures corresponds to the domain solved by one processor. The entire grid system was split into a 20 processor configuration to correspond with a 20 processor Beowulf Cluster.

The grid systems were designed to both include as much of the land area to the west of the lagoon as practically possible for simulation of flooding due to storms and include as much of the offshore area as possible to minimize the effect of the boundary on model results. Figure 7.6 compares the resolution of the CH3D IRL fine grid mentioned earlier(Figure 2.7) and the parallel PEM grids.

Using bathymetry and a shoreline provided by the SJRWMD (NAVD88), and the SJRWMD five foot Digital Elevation Model (DEM) topographic contours (NGVD29) of the IRL, the coarse (Figure 7.7) and fine grid bathymetry were developed. The shoreline was fixed to an elevation of 2.5 ft (NGVD29) and then combined with the DEM topographic data. The combined dataset was then converted to NAVD88 via Vertcon. Using the Florida Inland Navigation District (FIND) Survey 96 of the Atlantic Intracoastal Waterway (ICW), an ICW was imposed into both grid systems at a depth of 4 m (NAVD88).

Although neither hurricane made landfall in the IRL, the center of Hurricane Irene did pass through the parallel grid systems (Figure 7.8).

7.3.2 Alternative Possible Grid Systems

Currently, the PEM only allows the grid system to split entirely in the x-direction or entirely in the y-direction. However, assuming an overall grid size of 1400 × 640, many other possible domain decomposition configurations exist (Figure 7.9). Configuration A is used for this study.
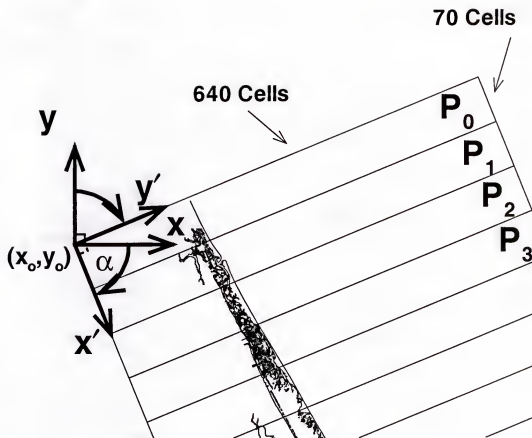
Figure 7.4: The Cartesian fine grid system is rotated α degrees about the point (x_o, y_o) where x_o is 480832 (UTM), y_o is 3.219e6 (UTM), and α is 67.7.

In general, the computational speed of the PEM is determined by how many messages must be sent between processors. While the configuration used in this study, A, was relatively easy to implement, it would not be the fastest because more messages are sent than several of the other configurations (Table 7.4). Configurations B and C require the fewest number of elements to be passed between processors. Thus, if the PEM were capable of splitting the domain in both the x- and y-directions simultaneously, Configurations B and C would be a better choice. As shown in Figure 7.9, Configurations B and C are also the configurations with aspect ratios closest to 1.
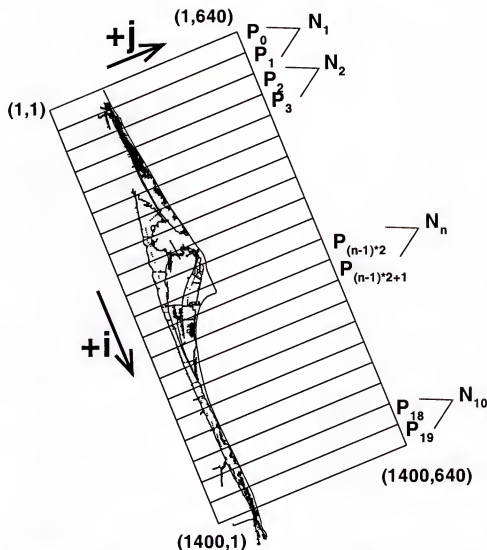
Figure 7.5: The IRL fine grid system developed for the PEM ($\Delta x = \Delta y = 125m$).

7.3.3 Boundary Conditions

This sections describes the development of the tidal, wind stress and pressure gradient boundary conditions using for PEM storm surge simulations.

Tidal forcing

Normally when simulating circulation and transport in an estuary exposed to the open ocean, using measured water level as a tidal boundary condition can be properly justified. However when trying to simulate a storm surge, measured water
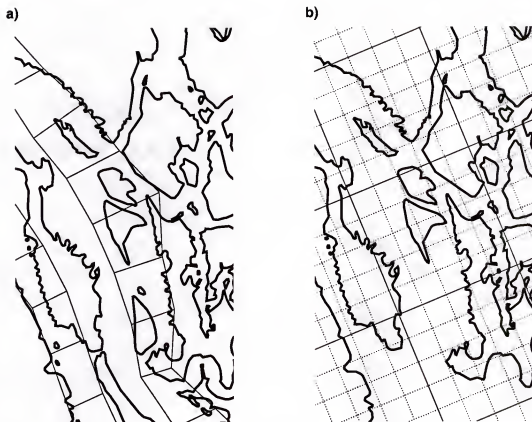
Figure 7.6: A resolution comparison between a) the IRL fine grid developed for CH3D (thin lines) and b) the IRL coarse (solid thin lines) and fine grids (dotted lines) developed for the PEM in the vicinity of Mosquito Lagoon.

Table 7.4: Number of elements sent by each processor to its corresponding neighbor processors for different 20 processor configurations of the overall fine grid (1400×640). The number of processors sending data in a particular direction is given in parenthesis. The number of cells ghost cells sent to a neighbor processor, T, is 2.

Configuration	m	n	N or S ($\frac{N_x}{m}T$)	E or W ($\frac{N_y}{n}T$)	NE,NW, SE,or SW (T^2)	Maximum (per processor)	Total (all processors)
A	20	1		1280(19)		2560	48640
B	10	2	280(10)	640(18)	4(9)	1568	28784
C	5	4	560(15)	320(16)	4(12)	1776	27232
D	1	20	2800(19)			5600	106400
E	2	10	1400(18)	128(10)	4(9)	2936	53104
F	4	5	700(16)	256(15)	4(12)	1928	30272
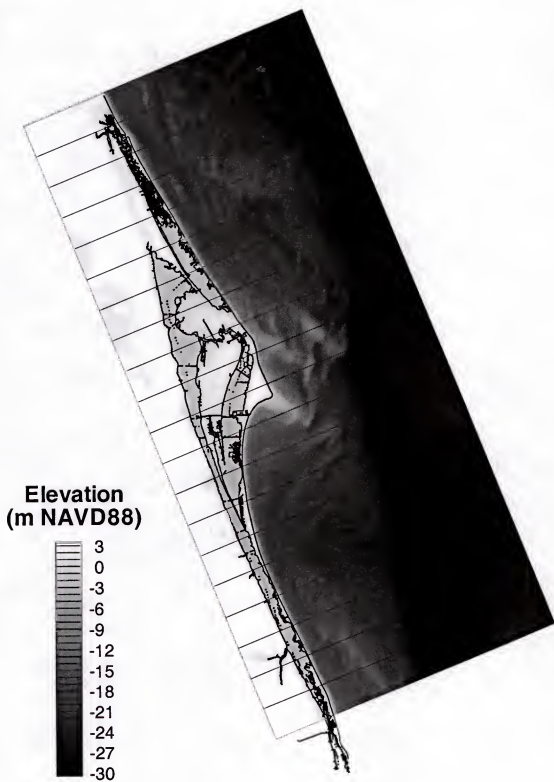
Figure 7.7: The elevation of the coarse grid system.
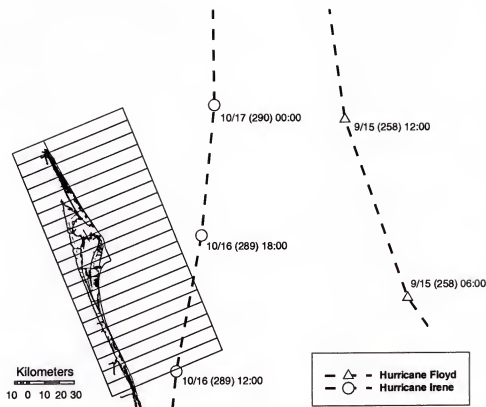
Figure 7.8: Tracks of Hurricanes Floyd and Irene (1999) in the vicinity of the 20 processor IRL PEM grids.

level cannot be used directly as tidal forcing because the measured water level data already contain the storm induced setup which the model is trying to simulate.

Instead, for the simulations presented herein, the tidal boundary condition was generated by combining a harmonically regenerated water level with a measured water level passed through a filter designed to remove tidal signals. This method of combining water levels allows both the tidal constituents and long term seasonal water level changes to be included in the tidal forcing. Before combining, the hurricane surges in the filtered water level were removed such that after combining, the water level should be equal to the measured water level at Sebastian Inlet except with the surges of Hurricane Floyd and Irene removed. The exact procedure for generating the tidal boundary condition appears below:
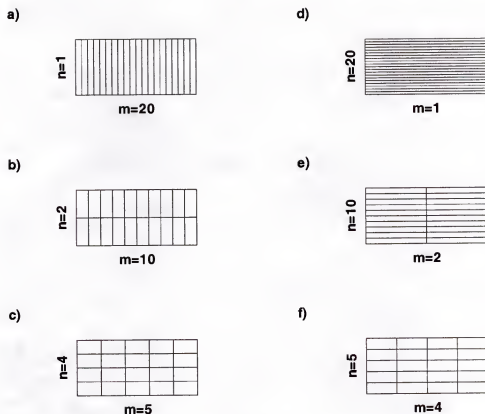
Figure 7.9: Possible 20 processor IRL parallel grid configurations. Assuming an overall grid size of 1400×640, the sub-grids would be of size a) 70×640 (0.11), b) 140×320 (0.44), c) 280×160 (1.75), d) 1400×32 (43.75), e) 700×64 (10.94), and f) 350×128 (2.73) where aspect ratios of the sub-grids are given in parenthesis.

- The measured water level station at Sebastian Inlet (FDEP #872-2004) (Figure 4.1) was selected as the base water level for the tidal boundary condition.

- A harmonic analysis was performed on the measured water level at Sebastian Inlet using the Institute of Ocean Sciences Tidal Package (IOS TP) (Foreman, 1996).

- Using the 60 constituents generated by the IOS TP, the tidal signal at Sebastian Inlet was regenerated.
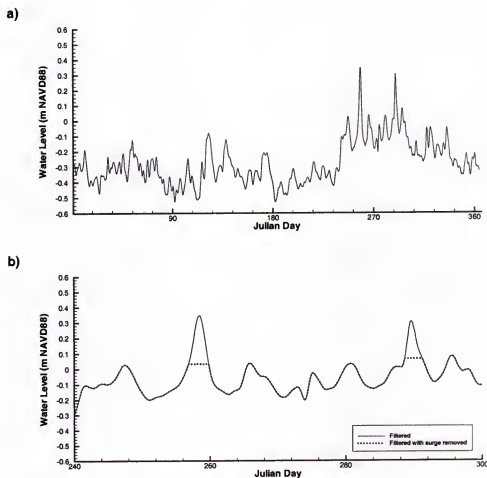
Figure 7.10: The filtered measured water level at Sebastian Inlet (FDEP #872-2004) a) during all of 1999 and b) during the passing of Hurricanes Floyd and Irene which also shows how the surges were removed.

- The Doodson and Warburg (1941) 39-hourly weighted average tidal filter as described by Groves (1955) was applied to the original Sebastian Inlet measured water level.

- To remove any high frequency noise in the filtered water level, another 5-hourly weighted average binomial filter (Panofsky and Brier, 1963) was applied. The twice filtered water level at Sebastian Inlet for 1999 appears in Figure 7.10a. The two large peaks on either side of Julian Day 270 are Hurricanes Floyd and Irene.
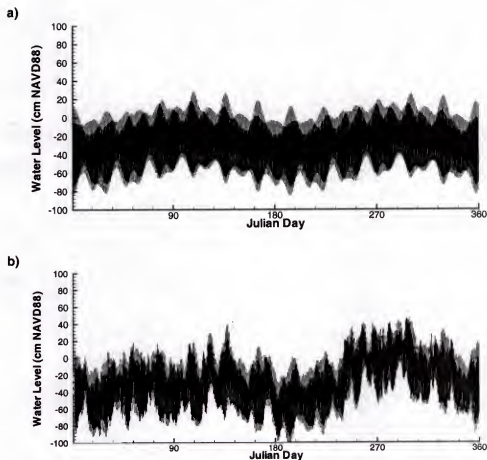
Figure 7.11: The a) harmonically regenerated water level at Sebastian Inlet and the b) combined filtered and harmonically regenerated water level at Sebastian Inlet.

- The Hurricane Floyd and Irene surges in the filtered water level data were cutoff (Figure 7.10b).

- The filtered water level with surges removed was combined with the de-meaned harmonically regenerated water level yielding the tidal boundary condition used by the model for the Hurricane Floyd and Irene simulations. The harmonically regenerated water level and the combined water level for 1999 appear in Figure 7.11.

<u>Atmospheric pressure gradient</u>

The wind model uses an exponentially decaying atmospheric pressure. The actual atmospheric pressure is not applied directly inside the model domain; however, the gradient in atmospheric pressure is applied to the u- and v-momentum equations. Using the equations described previously and the location, speed, direction and central pressure for a hurricane, the gradient of atmospheric pressure is applied. In addition, along open boundaries a inverse barometric effect term is applied.

<u>Wind forcing</u>

Using the wind model described in Chapter 5, a time and space varying hurricane wind field was created. This wind field is then converted to wind stress and applied directly to the u- and v-momentum equations. Simulations are performed using both constant (20 n mi) and measured radius of maximum wind speed as well as with several values for the ratio of surface wind velocity to gradient wind velocity, K (0.63, 0.83, 1). Comparisons between the measured wind speed and direction and the simulated wind speed and direction at Ponce de Leon Inlet (FDEP #872-1147), Titusville (FDEP #872-1456), Banana River (FDEP #872-1789) and Ft. Pierce Inlet (FDEP #872-2213) (Figure 4.1) during the passage of Hurricane Floyd using a constant radius of maximum wind speed and the measured radius of maximum wind speed are shown in Figures 7.12 and 7.13, respectively. Similar comparisons for Hurricane Irene are shown in Figures 7.14 and 7.15, respectively. Overall, the simulated and measured wind compare reasonably well for a simple wind model; however, some differences in angle and magnitude can be seen particularly in Titusville where the measurement station is well inland. Only small differences can be seen between the wind speed and direction simulated using a constant radius to maximum wind speed and the wind speed and direction simulated using a measured radius to maximum wind speed.

Figure 7.12: A comparison between the measured wind speed and direction (dotted lines) and the simulated wind speed and direction (solid lines) during the passage of Hurricane Floyd using the storm model at a) Ponce de Leon Inlet (FDEP #872-1147), b) Titusville (FDEP #872-1456), c) Banana River (FDEP #872-1789) and d) Ft. Pierce Inlet (FDEP #872-2213). The ratio of surface wind velocity to gradient wind velocity, K, used in the storm model is 0.63 and the radius of maximum wind speed is a constant 20 n mi

192

Figure 7.13: A comparison between the measured wind speed and direction (dotted lines) and the simulated wind speed and direction (solid lines) during the passage of Hurricane Floyd using the storm model at a) Ponce de Leon Inlet (FDEP #872-1147), b) Titusville (FDEP #872-1456), c) Banana River (FDEP #872-1789) and d) Ft. Pierce Inlet (FDEP #872-2213). The ratio of surface wind velocity to gradient wind velocity, K, used in the storm model is 0.63 and the radius of maximum wind speed is measured.

a)

b)

c)

d)
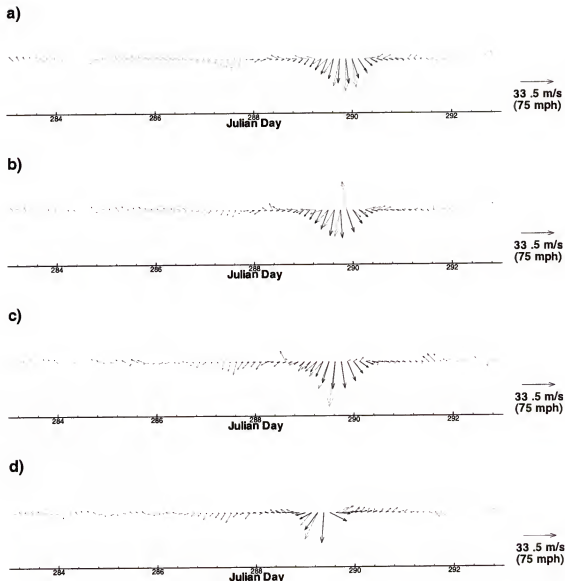
Figure 7.14: A comparison between the measured wind speed and direction (dotted lines) and the simulated wind speed and direction (solid lines) during the passage of Hurricane Irene using the storm model at a) Ponce de Leon Inlet (FDEP #872-1147), b) Titusville (FDEP #872-1456), c) Banana River (FDEP #872-1789) and d) Ft. Pierce Inlet (FDEP #872-2213). The ratio of surface wind velocity to gradient wind velocity, K, used in the storm model is 0.63 and the radius to maximum wind speed is a constant 20 n mi

a)

b)

c)

d)
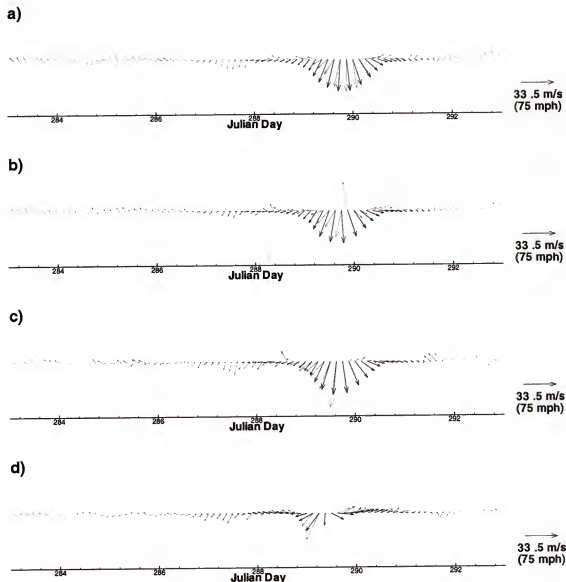
Figure 7.15: A comparison between the measured wind speed and direction (dotted lines) and the simulated wind speed and direction (solid lines) during the passage of Hurricane Irene using the storm model at a) Ponce de Leon Inlet (FDEP #872-1147), b) Titusville (FDEP #872-1456), c) Banana River (FDEP #872-1789) and d) Ft. Pierce Inlet (FDEP #872-2213). The ratio of surface wind velocity to gradient wind velocity, K, used in the storm model is 0.63 and the radius to maximum wind speed is measured.

7.4.1   Hurricane Floyd

Fourteen Hurricane Floyd simulations were performed using the PEM (Table 7.5). These simulations were designed to test the effect of grid spacing (625 m and 125 m), the ratio of surface wind velocity to gradient wind velocity (0.63, 0.83 and 1) and the type of radius to maximum winds (a constant 20 n mi or the measured values as shown in Table 7.2) on the simulated surge.

Table 7.5: Descriptions of the Hurricane Floyd simulations. The simulation periods are from September 10, 1999 (Julian Day 253) to September 19, 1999 (Julian Day 262) with all simulations using the measured position and translational velocity. K is the ratio of surface wind velocity to gradient wind velocity and the measured maximum wind speed is 135 mph.

Simulation Number	$\Delta x, \Delta y$ (m)	Radius to Maximum Winds (n mi)	K	Simulated Maximum Wind Speed (mph)
1	625	No Storm		
2	625	20	1	122.6
3	625	20	0.83	103.1
4	625	20	0.63	80.2
5	625	Measured	1	122.6
6	625	Measured	0.83	103.1
7	625	Measured	0.63	80.2
8	125	No Storm		
9	125	20	1	122.6
10	125	20	0.83	103.1
11	125	20	0.63	80.2
12	125	Measured	1	122.6
13	125	Measured	0.83	103.1
14	125	Measured	0.63	80.2

Because of the simplicity of the storm model coupled with Hurricane Floyd being considerably offshore, the simulated maximum wind speed never reached the measured 135 mph. Comparisons between time series of simulated and measured water level at three FDEP stations, Banana River (FDEP #872-1789), Melbourne

Causeway (FDEP #872-1843) and Sebastian Inlet (FDEP #872-2004) (Figure 4.1) are shown in Appendix P. The simulated water levels at the Banana River and Melbourne Causeway Stations show too much tidal influence suggesting that the grid with the 125 m grid spacing is a better choice for simulations. Comparisons between the simulated and measured water level at the Banana River and Melbourne Causeway Stations show that better results are obtained with a grid spacing of 125 m, the measured radius to maximum winds and a ratio of surface wind velocity to gradient wind velocity of 1, while at the Sebastian Inlet Station, a ratio of surface wind velocity to gradient wind velocity of 0.63 yields better results.

Conservation error was checked in all simulation as was always less than 0.1%. Figure 7.16 shows the typical amount of conservation error during several simulations in the coarse grid.

7.4.2 Hurricane Irene

Fourteen Hurricane Irene simulations were performed using the PEM (Table 7.6). As with the Hurricane Floyd simulations, these simulations were designed to test the effect of grid spacing (625 m and 125 m), the ratio of surface wind velocity to gradient wind velocity (0.63, 0.83 and 1) and the type of radius to maximum winds (a constant 20 n mi or the measured values as shown in Table 7.3) on the simulated surge.

For the Hurricane Irene simulations, the PEM was able to simulate the maximum wind speed quite well (with a ratio of surface wind speed to gradient wind speed between 0.63 and 0.83).

Time series comparisons between simulated and measured water level at three FDEP stations, Banana River (FDEP #872-1789), Melbourne Causeway (FDEP #872-1843) and Sebastian Inlet (FDEP #872-2004) (Figure 4.1) are shown in Appendix Q. The simulated water levels at the Banana River and Melbourne Causeway Stations again show too much tidal influence suggesting that the grid with the 125 m
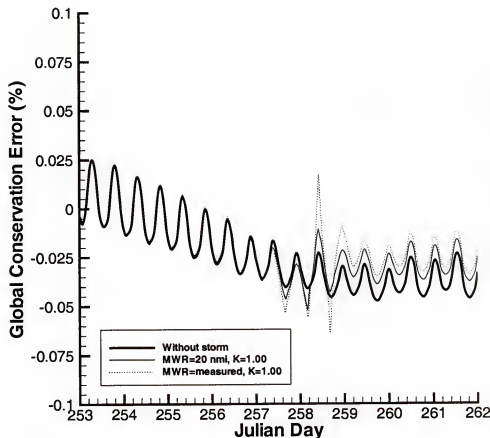
Figure 7.16: Conservation error during simulations of the passage of Hurricane Floyd. The error occurs during Simulations 2 and 5 in the coarse grid.

grid spacing is a better choice for simulations. Comparisons between the simulated and measured water level at the Banana River, Melbourne Causeway and Sebastian Inlet show that a grid spacing of 125 m, the measured radius to maximum winds and a ratio of surface to gradient wind velocity of 1 generate the most accurate surges.

Again, conservation error was checked in all simulation as is was always less than 0.1%. Figure 7.17 shows the typical amount of conservation error during several simulations in the coarse grid.

7.5 Hypothetical Hurricane

The final series of simulations are designed to study the effect of Hurricanes making direct contact with the IRL. Using the historically most probable approach

Table 7.6: Descriptions of the Hurricane Irene simulations. The simulation periods are from October 13, 1999 (Julian Day 286) to October 19, 1999 (Julian Day 292) with all simulations using the measured position and translational velocity. K is the ratio of surface wind velocity to gradient wind velocity and the measured maximum wind speed is 95 mph.

Simulation Number	$\Delta x, \Delta y$ (m)	Radius to Maximum Winds (n mi)	K	Simulated Maximum Wind Speed (mph)
1	625	No Storm		
2	625	20	1	112.7
3	625	20	0.83	99.0
4	625	20	0.63	83.0
5	625	Measured	1	112.7
6	625	Measured	0.83	99.0
7	625	Measured	0.63	83.0
8	125	No Storm		
9	125	20	1	112.7
10	125	20	0.83	99.0
11	125	20	0.63	83.0
12	125	Measured	1	112.7
13	125	Measured	0.83	99.0
14	125	Measured	0.63	83.0

direction of a landfalling hurricane determined by Dean and Chiu (1986). (Figure 7.18), 130°, a hypothetical hurricane track was developed (Figure 7.19). Storms following this track make landfall just north of Cape Canaveral.

Using this hurricane track, 21 hypothetical hurricane simulations were performed using the PEM (Table 7.7). These simulations were designed to study how the simulated flooding would be effected by large storms making landfall on the IRL under a variety of grid spacings (625 m and 125 m), tidal phases at the time of landfall (Figure 7.20) and ratios of surface wind velocity to gradient wind velocity (0.63, 0.83, and 1). The time period of all the simulations was from October 3, 1999 to October 9, 1999. The same combined harmonically regenerated and filtered water level
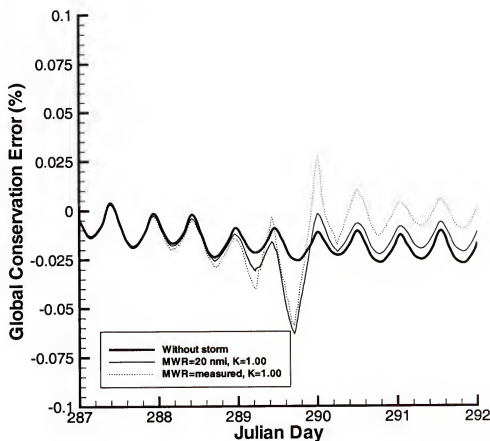
Figure 7.17: Conservation error during simulations of the passage of Hurricane Irene. The error occurs during Simulations 2 and 5 in the coarse grid.

used for the Hurricane Floyd and Irene simulations were used as the tidal boundary condition for the hypothetical hurricane simulations.

Figures 7.21 (Cases 2 through 7) and 7.22 (Cases 12-17) compare the flooding caused by a Category 2 hurricane making landfall at different phases of the tidal cycle in the IRL PEM coarse and fine grids, respectively. The simulations show more flooding on the lagoon side of Merritt Island (West) rather than on the ocean side (East) and considerably more flooding in the coarse grid as opposed to the fine grid. More flooding occurs on the west side because of its relatively low topography while more flooding occurs in the coarse grid because its resolution does not allow for proper representation of the topography. Additionally, the simulated maximum
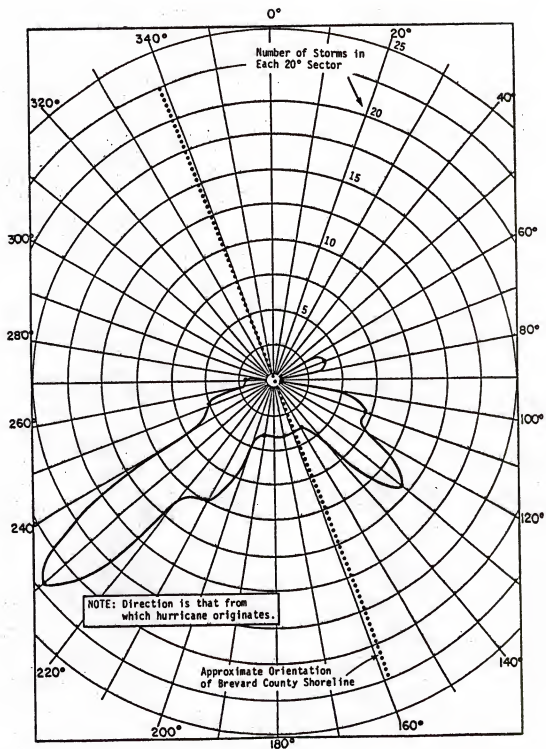
Figure 7.18: The directional distribution of historical storms at the mid-location of the Brevard County coastline (Dean and Chiu, 1986).
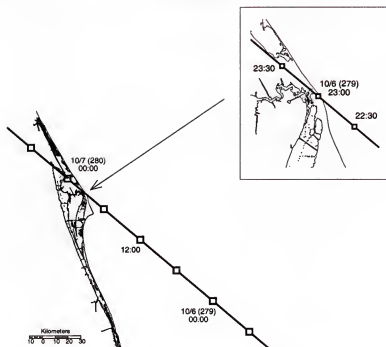
Figure 7.19: The track of the hypothetical hurricane making landfall at False Cape, just north of Cape Canaveral, in the IRL during October 1999.

flooding does not occur when the storm makes landfall at high tide, but rather when the storm makes landfall several hours after high tide.

Figures 7.23 (Cases 9, 5 and 10) and 7.24 (Cases 19, 15 and 20) compare the amount of flooding due to hypothetical Category 1, 2 and 3 hurricanes making landfall at high tide in the coarse and fine grids, respectively. The fine grid shows less flooding than the coarse grid because of the more accurate fine grid bathymetry.

Figure 7.25 (Case 21) illustrates the amount of flooding caused by a major Category 5 hurricane making landfall at high tide. This simulation shows the large amount of flooding which would be associated with landfall of a major hurricane. All of the northern part of Merritt Island is flooded along with a significant portion of the river front in the northern IRL. As with the previous simulations, more flooding is seen on the IRL itself rather than areas closer to the ocean such as Banana River.

Table 7.7: Descriptions of the hypothetical hurricane simulations. The simulation periods are from October 3, 1999 (Julian Day 276) to October 9, 1999 (Julian Day 282). All simulations use hypothetical hurricanes with approach angles of 130° from the North, translational velocities of 10 knots and a radius to maximum wind speed of 20 n mi Simulated hurricanes make landfall at longitude 80 34 34 W (UTM 541468) and latitude 28 35 12 N (UTM 3162303). The time of high tide, T_H, is 22:57, October 6, 1999 (Julian Day 279.956) and K is the ratio of surface wind velocity to gradient wind velocity.

Simulation Number	$\Delta x, \Delta y$ (m)	Landfall Time	ΔP (in hg)	K	Maximum Wind Speed (mph)
1	625	No Storm			
2	625	$T_H - 6$ hrs	-2.0	0.9	94.1
3	625	$T_H - 4$ hrs	-2.0	0.9	94.1
4	625	$T_H - 2$ hrs	-2.0	0.9	94.1
5	625	T_H	-2.0	0.9	94.1
6	625	$T_H + 2$ hrs	-2.0	0.9	94.1
7	625	$T_H + 4$ hrs	-2.0	0.9	94.1
8	625	$T_H + 6$ hrs	-2.0	0.9	94.1
9	625	T_H	-2.0	0.7	74.5
(5)	625	T_H	-2.0	0.9	94.1
10	625	T_H	-2.0	1.1	113.8
11	125	No Storm			
12	125	$T_H - 6$ hrs	-2.0	0.9	94.1
13	125	$T_H - 4$ hrs	-2.0	0.9	94.1
14	125	$T_H - 2$ hrs	-2.0	0.9	94.1
15	125	T_H	-2.0	0.9	94.1
16	125	$T_H + 2$ hrs	-2.0	0.9	94.1
17	125	$T_H + 4$ hrs	-2.0	0.9	94.1
18	125	$T_H + 6$ hrs	-2.0	0.9	94.1
19	125	T_H	-2.0	0.7	74.5
(15)	125	T_H	-2.0	0.9	94.1
20	125	T_H	-2.0	1.1	113.8
21	125	T_H	-2.5	1.4	159.5

7.6 Simulation CPU Times

The average CPU time per iteration for all of the aforementioned hurricane simulations is shown in Table 7.8. The fine grid, which has 25 times more computational cells than the coarse grid, is approximately 43 times slower than the coarse
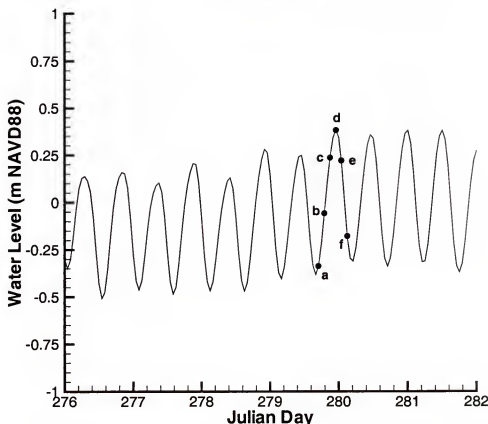
Figure 7.20: The measured water level at the Sebastian Inlet Station (FDEP #872-2004) which is used as tidal forcing during hypothetical hurricane simulations. The letters: a, b, c, d, e, and f indicate the water level at the time of landfall: 6 hours before high tide, 4 hours before high tide, 2 hours before high tide, high tide, 2 hours after high tide and 4 hours after high tide.

grid for simulations of Hurricane Floyd and Irene and 22 times slower for hypothetical hurricane simulations. The differences in the CPU times were attributed to the nature of the simulations. During a typical simulation, approximately 97% of the computational time was spent solving the 9-diagonal matrix for water level and calculating the non-linear and diffusion terms using the parallel ELM algorithm. The CPU time for the matrix solver varies because it is an iterative solver which implies that the time necessary to solve a matrix is a function of how much the water level
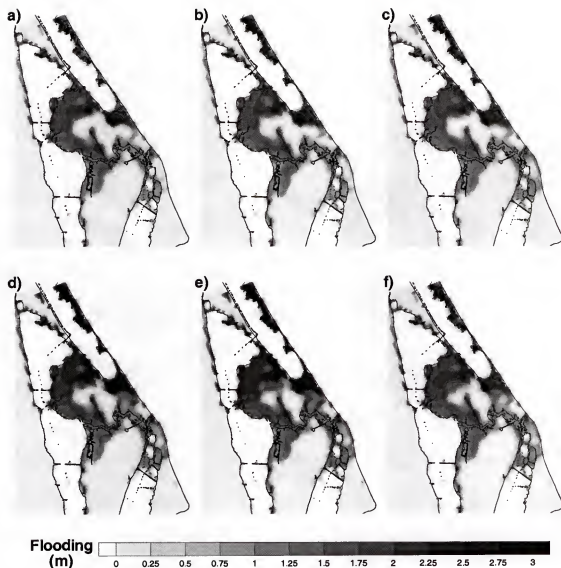
Figure 7.21: The simulated maximum depth of flooding during the passage of hypothetical Category 2 hurricanes using the coarse grid. The hurricane makes landfall at high tide a) −6 hours (391 km^2/1.28 m), b) −4 hours (397 km^2/1.30 m), c) −2 hours (400 km^2/1.37 m), d) +0 hours (413 km^2/1.49 m), e) +2 hours (430 km^2/1.52 m) and f) +4 hours (421 km^2/1.41 m). The surface area of flooded land and average depth of flooding are shown in parenthesis

changes during a time step. The CPU time for the parallel ELM also varied during the simulations because of differences in the time necessary to track back the velocities.
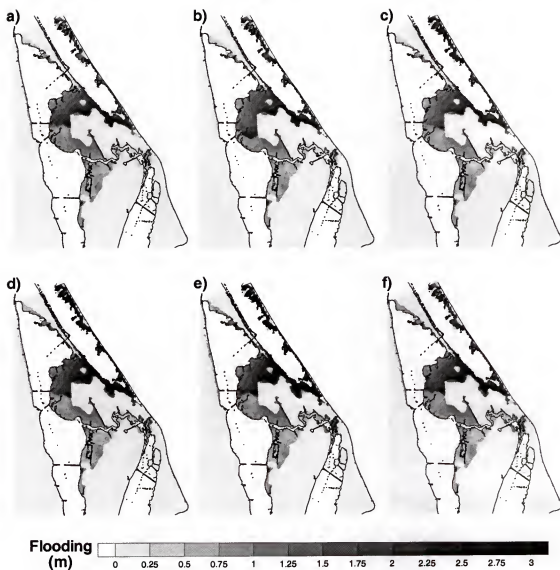
Figure 7.22: The simulated maximum depth of flooding during the passage of hypothetical Category 2 hurricanes using the fine grid. The hurricane makes landfall at high tide a) −6 hours (229 km²/1.01 m), b) −4 hours (226 km²/1.08 m), c) −2 hours (233 km²/1.14 m), d) +0 hours (245 km²/1.23 m), e) +2 hours (254 km²/1.29 m) and f) +4 hours (247 km²/1.18 m). The surface area of flooded land and average depth of flooding are shown in parenthesis
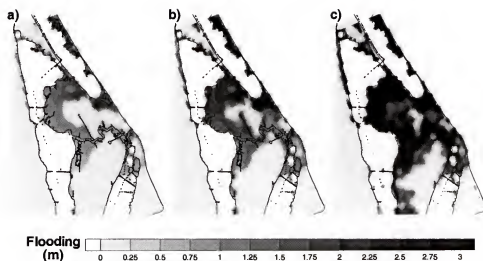
Figure 7.23: The simulated maximum depth of flooding during the passage of hypothetical a) Category 1 (313 km²/1.01 m), b) Category 2 (413 km²/1.49 m), and c) Category 3 (562 km²/2.15 m) hurricanes using the coarse grid and making landfall at high tide. The surface area of flooded land and average depth of flooding are shown in parenthesis
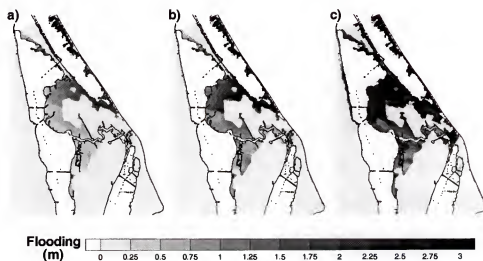
Figure 7.24: The simulated maximum depth of flooding during the passage of hypothetical a) Category 1 (196 km²/0.78 m), b) Category 2 (245 km²/1.23 m), and c) Category 3 (313 km²/2.07 m) hurricanes using the fine grid and making landfall at high tide. The surface area of flooded land and average depth of flooding are shown in parenthesis
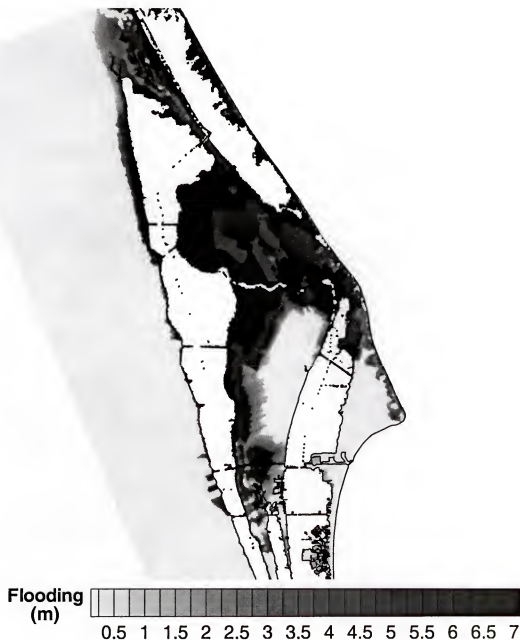
Figure 7.25: The simulated maximum depth of flooding during the passage of hypothetical Category 5 hurricane using the fine grid and making landfall at high tide (584 km^2/5.04 m). The surface area of flooded land and average depth of flooding are shown in parenthesis

Table 7.8: PEM simulation CPU times for the simulations of Hurricanes Floyd and Irene and the hypothetical hurricane. Standard deviations of the average CPU time for the various cases are shown in parenthesis.

Hurricane	Average CPU time per iteration in the		$\times$ faster than real time in the	
	Coarse Grid	Fine Grid	Coarse Grid	Fine Grid
Floyd	2.09(0.02)	92.01(4.67)	430.62	9.78
Irene	1.96(0.04)	85.96(5.51)	459.18	10.47
Hypothetical	3.90(0.14)	82.22(4.86)	229.67	10.98

CHAPTER 8
SUMMARY AND CONCLUSIONS

8.1 Conclusions

This study sought to explore parallel techniques to create high performance numerical models of estuarine circulation and transport. Two parallel models were developed: a legacy serial numerical model, CH3D, was converted to a parallel model by wrapping sections of the code with parallel constructs and an entirely new model was created. A parallel shared memory version of the CH3D hydrodynamic, sediment and nutrient model was created. The parallel CH3D hydrodynamic and salinity models were then applied to a year-long study of the IRL. Comparisons between simulated water level, flow rate and salinity and measured data showed good agreement. Additionally, a new parallel wetting and drying model was developed, tested and applied to storm surge simulations in the IRL. Overall, the following accomplishments have been achieved:

- The governing equations of the CH3D model were reviewed and a coarse and fine grid system were developed for study of the IRL.

- Based on the independent nature of the sweeping schemes used in the CH3D model, a new parallel CH3D model was created by wrapping the sweeps' computational loops in macros consisting of parallel constructs. Several parallel grid systems for the IRL were then generated based on several domain decomposition techniques. The most efficient grid system divided the row weight as evenly as possible among each of the processing elements.

- An overview of measured data available within the IRL was presented. The measured inlet water level data were used as tidal forcing; however, some adjustments were necessary because the water level gauges were not located exactly on the grid boundary. The mean water level at Ponce de Leon Inlet was also reduced to prevent an artificially large south to north setup from forcing large amounts of saline water out of Mosquito Lagoon through Haulover Canal. A series of year-long sensitivity tests were then performed to determine which model parameters were most important. Overall, the model was fairly insensitive to bottom roughness and the horizontal diffusion coefficient and although eight vertical layers produced slightly better results, the doubling of the execution time far outweighed any slight benefit in accuracy.

- The newly developed parallel CH3D model was used to perform one year hydrodynamic and salinity simulations of the IRL. The difference between the simulated and measured mean water level was 3.4% and the RMS error between the simulated and measured water level and the simulated and measured salinity was 5.3% and 18.1%, respectively. For the one year IRL simulations, the parallel CH3D model was shown to be conservative and produced the same results as the serial CH3D model. Speedup and efficiency of the newly developed parallel CH3D model were determined using a series of five day simulations under three different computing environments. The fine grid simulation of hydrodynamics, salinity, sediment and water quality was shown to have the best speedup. On "ocean", using two, three, and four processors, speedups of 1.82, 2.45 and 3.01 were attained with corresponding efficiencies of 91%, 82% and 75%.

- A new parallel hydrodynamic numerical model was developed. The hydrodynamic model includes implicit calculation of Coriolis, a parallel ELM routine to

solve the non-linear and diffusion terms, and a coupled storm model which can be used to simulate the surge associated with large tropical storms. The model was tested and verified using several analytical tests. For large grid systems, a speedup of 18 was achieved using an explicit model formulation and a speedup of 15 was achieved using a semi-implicit formulation.

- Coarse and fine grid systems were developed for study of the IRL using the PEM. Using these grid systems, the surge associated with the passage of Hurricanes Floyd and Irene were calculated and compared with the measured water level at several stations in the lagoon. These simulations showed the model's ability to simulate water level setup reasonably well especially when the fine grid is used; however, the comparisons between simulated and measured wind speed and direction illustrate the inadequacies of the simple storm model. Simulations of hypothetical hurricanes impacting False Cape showed the maximum amount of flooding occurs when the hurricane makes landfall several hours after high tide. A Category 5 hurricane making landfall was shown to produce flooding of up to 7 m.

8.2 Discussion and Recommendations

Although the numerical models developed and applied in this study have performed well simulating water level, flow rate and salinity transport in the IRL, the models contains various uncertainties, assumptions and simplifications which need further investigation and improvement.

8.2.1 CH3D Recommendations

Several possible improvements in the developed parallel CH3D model and the year-long simulations of circulation and transport in the IRL appear in the following sections.

The developed parallel CH3D model

- **Use larger number of processors.** Parallel simulations could be performed on shared memory computers with more than four processors to further examine the speedup properties of the parallel CH3D model.

- **Use finer grid systems.** Finer grid systems could be created to study how the parallel efficiency improves as the number of calculations increase.

- **Combine transport routines.** The transport routines (salinity, sediment, species, nutrient) could all be combined into one routine which calculates all transport variables simultaneously. By solving all transport variables within the same loops, serial overhead would be reduced thus increasing the parallel efficiency.

1-year IRL hydrodynamics and transport simulation

- **Generate a finer grid system.** By generating a finer grid system, the numerous small islands located in Mosquito Lagoon can be resolved which is necessary to fully understand the tidal superelevation in the vicinity of Ponce de Leon Inlet.

- **Obtain more measured data.** Although the measured data in IRL were extensive, certain types of data to be used for model comparison were unavailable. For example, measured discharge data in the northern IRL and Mosquito Lagoon, measured evaporation rate, long-term flow through causeways, current data, additional salinity gauges placed far away from the causeways and long-term salinity data at the inlets.

- **Expand the IRL grid systems northward.** By expanding the grid systems northward, the entire Halifax River could be included into the model domain

which would allow easier specification of the northern water level and salinity boundary conditions.

- **Expand the IRL grid systems to include offshore.** Including the offshore region into the grid systems would allow for better specification of both water level and salinity boundary conditions.

8.2.2   PEM Recommendations

Several possible improvements in the newly developed PEM model and the simulations of hurricanes impacting the IRL appear in the following sections.

Model capabilities

- **Change the model from 2-d to 3-d.** Fully 3-d models can better calculate bottom friction than vertically averaged 2-d models because velocities in the layer closest to the bottom are used instead of the vertically averaged velocities. Additionally, 3-d models are better suited to calculate salinity transport than 2-d models.

- **Add conservative species transport.** Generic species transport would allow for flushing simulations, salinity and temperature transport could be used to calculate baroclinic forcing, and sediment and nutrient transport would allow the addition of a water quality model.

- **Improve the storm model.** With the addition of a more robust storm model, *e.g.* a planetary boundary layer model (PBL), a more accurate wind and pressure gradient field would be created.

- **Boundary-fitted grid system**. By modifying the grid system and governing equations, a boundary-fitted grid system could be developed as opposed to the current grid system which resolves shorelines in a stair-step fashion.

Hurricane storm surge simulations

- **Obtain additional bathymetric and topographic data.** Due to inadequacies of the current bathymetric data, high density bathymetric data is needed near selected causeways and tidal inlets. The only topographic data available in the IRL comes from the Digital Elevation Model's five foot contours so there is no topographic resolution between zero and five feet.

- **Create spatially varying bottom friction coefficients.** Currently, the PEM uses a constant Manning's n in all grid cells. A constant value over the entire domain is not appropriate especially for the areas of land flooded by the storm surge.

- **Link the model with a Geographic Information Systems (GIS) database.** Based on the success of linking CH3D with GIS, the PEM could also be linked with GIS and used to aid in the visualization of simulated storm surge in the IRL. In addition, using a GIS land use database, emergency managers could predict which municipalities will be most effected by flooding including specific details such as the locations of impassable roads.

- **Additional surge simulations.** The model could be used to simulate the flooding due to various combinations of storm intensity, approach direction, translational velocity and landfall locations. By simulating a number of different scenarios, an approaching storm could be categorized into a specific scenario and the appropriate flooding map could be used to issue evacuation orders.

- **Generate a finer grid system.** A fine grid system would better resolve small features and improve estimates of local flooding.

- **Obtained measured surge data.** To determine if the model correctly predicts the flooding associated with landfall of a hurricane it is desirable to have storm surge gauges placed at various locations in the lagoon.

APPENDIX A
CH3D HORIZONTAL DIFFUSION TERMS

The curvilinear coordinate, horizontal diffusion terms used by CH3D in the $u-$ velocity equations are

$$
\begin{aligned}
& u_{11}H_{11} + 2u_{12}H_{12} + u_{22}H_{22} \\
+\ & (D^1_{11}u)_1H_{11} + (D^1_{11}u)_2H_{12} + (D^1_{12}u)_1H_{12} + (D^1_{12}u)_2H_{22} \\
+\ & (D^1_{12}v)_1H_{11} + (D^1_{12}v)_2H_{12} + (D^1_{22}v)_1H_{12} + (D^1_{22}v)_2H_{22} \\
-\ & (D^1_{12}H_{12} + D^2_{22}H_{22})u_1 \\
+\ & (D^1_{11}H_{12} + D^1_{12}H_{22} - D^2_{11}H_{11} - 2D^2_{12}H_{12} - D^2_{22}H_{22})u_2 \\
+\ & (D^1_{12}H_{11} + D^1_{22}H_{12})v_1 + (D^1_{12}H_{12} + D^1_{22}H_{22})v_2 \\
+\ & [(D^1_{12}D^1_{12} + D^2_{22}D^2_{12} - D^1_{22}D^1_{11} - D^2_{22}D^2_{12})H_{22} + (D^2_{22}D^2_{11} - D^2_{22}D^1_{12})H_{12}]\,u \\
+\ & [(D^1_{11}D^1_{22} + D^1_{12}D^2_{22} - D^2_{12}D^1_{22} - D^1_{22}D^1_{12})H_{12} + (D^1_{12}D^1_{22} - D^1_{11}D^1_{22})H_{11}]\,v
\end{aligned}
\tag{A.1}
$$

and the horizontal diffusion terms in the $v-$ velocity equations are

$$
\begin{aligned}
& v_{11}H_{11} + 2v_{12}H_{12} + v_{22}H_{22} \\
+\ & (D^2_{11}u)_1H_{11} + (D^2_{11}u)_2H_{12} + (D^2_{12}u)_1H_{12} + (D^2_{12}u)_2H_{22} \\
+\ & (D^2_{12}v)_1H_{11} + (D^2_{12}v)_2H_{12} + (D^2_{22}v)_1H_{12} + (D^2_{22}v)_2H_{22} \\
-\ & (D^2_{11}H_{11} + D^2_{12}H_{12})v_2 \\
+\ & (D^2_{22}H_{11} + D^2_{22}H_{12} - D^1_{11}H_{11} - 2D^1_{12}H_{11} - D^2_{22}H_{22})v_1 \\
+\ & (D^2_{11}H_{11} + D^2_{12}H_{12})u_1 + (D^2_{11}H_{12} + D^2_{12}H_{22})u_2 \\
+\ & [(D^1_{11}D^1_{12} + D^2_{11}D^1_{22} - D^1_{12}D^2_{11} - D^2_{22}D^2_{11})H_{12} + (D^2_{12}D^1_{11} - D^2_{22}D^1_{11})H_{22}]\,u \\
+\ & [(D^2_{12}D^1_{12} + D^2_{11}D^1_{11} - D^1_{11}D^1_{12} - D^2_{11}D^2_{22})H_{11} + (D^2_{11}D^1_{12} - D^1_{12}D^2_{12})H_{12}]\,v
\end{aligned}
\tag{A.2}
$$

where H_{ij} is the inverse metric tensor, D^i_{jk} is the Christoffel symbol of the second kind and the other subscripts represent partial differentiation.

APPENDIX B
TENSOR INVARIANT EQUATIONS OF MOTION

Following the derivations presented in Sokolnikoff (1960), the tensor invariant, 3-D equations of motion and transport can be derived in terms of the contravariant vector components. These equations are independent of the frame of reference chosen and are the more general form of the Cartesian equations presented earlier. They non-dimensional form can be written as

$$\zeta_t + \frac{\beta}{\sqrt{g_o}} \frac{\partial}{\partial x^k}(\sqrt{g_o} H u^k) \tag{B.1}$$

$$
\begin{aligned}
\frac{1}{H} \frac{\partial H u^k}{\partial t} = & -\zeta^{|k} - g_{nj}\epsilon^{kj} u^n \\
& - \frac{R_o}{H}\left[(H u^l u^k)_{,l} + \frac{\partial h u^k \omega}{\partial \sigma}\right] \\
& + \frac{E_v}{H^2}\frac{\partial}{\partial \sigma}\left(A_v \frac{\partial u^k}{\partial \sigma}\right) + E_h A_h u^k_{,m}{}^{|m} \\
& - \frac{R_o}{F_{r_D}^2}\left[H \int_\sigma^0 \rho^{|k}d\sigma + H^{|k}\left(\int_\sigma^0 \rho d\sigma + \sigma\rho\right)\right]
\end{aligned}
\tag{B.2}
$$

$$
\begin{aligned}
\frac{\partial S}{\partial t} + & \frac{R_o}{H}\left[(H u^l S)_{,l} + \frac{\partial h \omega S}{\partial \sigma}\right] \\
= & \frac{E_h}{Sc_h} D_h S_{,m}{}^{|m} + \frac{E_v}{Sc_v}\frac{1}{H^2}\frac{\partial}{\partial \sigma}\left(D_v \frac{\partial S}{\partial \sigma}\right)
\end{aligned}
\tag{B.3}
$$

where $\frac{\partial}{\partial x^k}$ is the partial derivative, g_{ij} is the metric tensor which in two dimensions is defined as

$$g_{ij} = \begin{pmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{pmatrix} = \begin{pmatrix} x_\xi^2 + y_\xi^2 & x_\xi x_\eta + y_\xi y_\eta \\ x_\eta x_\xi + y_\eta y_\xi & x_\eta^2 + y_\eta^2 \end{pmatrix} \tag{B.4}$$

g_o is the Jacobian and is equal to $x_\xi y_\eta - x_\eta y_\xi$, ϵ^{kj} is the permutation tensor such that

$$
\begin{aligned}
\epsilon^{12} &= -\frac{1}{\sqrt{g_o}} \\
\epsilon^{21} &= \frac{1}{\sqrt{g_o}} \\
\epsilon^{11} &= \epsilon^{22} = 0
\end{aligned}
\tag{B.5}
$$

Two other derivative operators are given in Equations B.1 through B.3, these are the covariant spatial derivative and the contravariant spatial derivative

$$u^i_{,j} = u^i_{;j} + D^i_{\alpha j} u^\alpha \tag{B.6}$$

$$S!^k = g^{km}S_{,m} \tag{B.7}$$

where ": j" represents partial differentiation and $D^i_{\alpha j}$ is the Christoffel symbol of the second kind

$$D^i_{\alpha j} = g^{in}D_{njk} \tag{B.8}$$

where g^{in} is the inverse metric tensor and D_{njk} is the Christoffel symbol of the first kind

$$D_{njk} = \frac{1}{2}\left(g_{ij;k} + g_{ik;j} + g_{jk;i}\right) \tag{B.9}$$

APPENDIX C
CH3D TURBULENCE MODEL

The starting point of a turbulence model is to derive the mean flow equations from the conservation equations for mass, momentum, energy and species. For incompressible flows, the conservation equations can be written in tensor notation as

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{C.1}$$

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho_o}\frac{\partial p}{\partial x_i} + g_i\frac{(\rho - \rho_o)}{\rho_o} - 2\epsilon_{ijk}\Omega_j u_k + \frac{\partial}{\partial x_j}\left(\nu\frac{\partial u_i}{\partial x_j}\right) \tag{C.2}$$

$$\frac{\partial \phi}{\partial t} + u_j\frac{\partial \phi}{\partial x_j} = \frac{\partial}{\partial x_j}\left(k\frac{\partial \phi}{\partial x_j}\right) \tag{C.3}$$

where ϕ is either temperature or salinity, ρ is density, ρ_o is a reference density, ϵ_{ijk} is the permutation tensor, Ω_j is the rotational speed of the earth and ν and k are the molecular viscosity and diffusivity.

The the variables are split into mean and fluctuating quantities, the mean flow equations can be written as

$$\frac{\partial \overline{u}_i}{\partial x_i} = 0 \tag{C.4}$$

$$\frac{\partial \overline{u}_i}{\partial t} + \overline{u}_j\frac{\partial \overline{u}_i}{\partial x_j} = -\frac{\partial \overline{u_i'u_j'}}{\partial x_j} - \frac{1}{\rho_o}\frac{\partial \overline{p}}{\partial x_i} + g_i\frac{(\rho - \rho_o)}{\rho_o} - 2\epsilon_{ijk}\Omega_j\overline{u}_k + \frac{\partial}{\partial x_j}\left(\nu\frac{\partial \overline{u}_i}{\partial x_j}\right) \tag{C.5}$$

$$\frac{\partial \overline{\phi}}{\partial t} + \overline{u}_j\frac{\partial \overline{\phi}}{\partial x_j} = -\frac{\partial \overline{u_j'\phi'}}{\partial x_j} + \frac{\partial}{\partial x_j}\left(k\frac{\partial \overline{\phi}}{\partial x_j}\right) \tag{C.6}$$

where $\overline{u}_i$, $\overline{u}_j$, and $\overline{u}_k$ are the mean velocity components, u_i' and u_j' are the fluctuating velocity components, $\overline{\phi}$ and $\overline{\phi'}$ are the mean and fluctuating temperature or salinity, and $\overline{p}$ is the mean pressure. This system of equations is not closed because of the Reynolds stress terms.

Turbulence models can be classified as eddy viscosity models and second-order closure models. The eddy viscosity models assume the turbulent Reynolds stresses to be the products of mean velocity gradients and "eddy viscosities" and the turbulent heat and mass fluxes to be the products of mean temperature and concentration gradients and "eddy diffusivities"

$$-\overline{u_i' u_j'} = A_t \left(\frac{\partial \overline{u}_i}{\partial x_j} + \frac{\partial \overline{u}_j}{\partial x_i} \right) - \frac{q^2}{3} \delta_{ij} \tag{C.7}$$

$$-\overline{u_i' \phi'} = K_t \frac{\partial \overline{\phi}}{\partial x_i} \tag{C.8}$$

where $q^2 = \overline{u_i' u_i'}$, and A_t and K_t are turbulent eddy viscosity and diffusivity, respectively.

The eddy coefficients are assumed to be known functions of, for example, mean flow parameters (Munk and Anderson, 1948), depths and wind speeds. These functions are difficult to determine and are often *ad-hoc* in nature and may have to be adjusted for application to a new site or a new flow situation.

The second-order closure models resolve the dynamics of turbulence by including the differential transport equations for the turbulence variables' *i.e.*, the second-order correlations, $-\overline{u_i' u_j'}$, $-\overline{u_i' \phi'}$, and $\overline{\phi' \phi'}$. In the most complicated case, *i.e.*, the Reynolds stress model, the differential transport equations are solved (Lewellen, 1977; Sheng, 1982; Sheng and Villaret, 1989).

An equilibrium closure model (Sheng, 1985; Sheng and Chiu, 1986; Sheng et al., 1990b) is used to compute the vertical turbulence. The local equilibrium condition is valid when the time scale of mean flow is much larger than that for turbulence and when turbulence varies little over the turbulence macroscale. In this case, the evolution and diffusion terms become negligible with respect to the other terms. The equilibrium close model is significantly simpler than the complete second-order closure model (Reynolds stress model) and has been found to give good results in mean flow, salinity and temperature with little or no tuning of coefficients.

In addition to the mean flow equations, a set of algebraic equations are solved for the second-order correlation quantities to obtain the stability functions ϕ_1 and ϕ_2 in terms of the mean flow variables. These equations, when written in dimensional and tensor forms, are

$$
\begin{aligned}
0 = \quad & - \overline{u_i'u_k'}\frac{\partial \overline{u}_j}{\partial x_k} - \overline{u_j'u_k'}\frac{\partial \overline{u}_i}{\partial x_k} + g_i\frac{\overline{u_j'\rho'}}{\rho_o} + g_j\frac{\overline{u_i'\rho'}}{\rho_o} \\
& - 2\epsilon_{ikl}\Omega_k\overline{u_l'u_j'} - 2\epsilon_{jlk}\Omega_l\overline{u_k'u_i'} - \frac{q}{\Lambda}\left(\overline{u_i'u_j'} - \delta_{ij}\frac{q^2}{3}\right) - \delta_{ij}\frac{q^3}{12\Lambda}
\end{aligned}
\tag{C.9}
$$

$$
0 = -\overline{u_i'u_j'}\frac{\partial \overline{\rho}}{\partial x_j} - \overline{u_j'\rho'}\frac{\partial \overline{u}_i}{\partial x_j} - \frac{g_i\overline{\rho'\rho'}}{\rho_o}
\tag{C.10}
$$

$$
0 = 2\overline{u_j'\rho'}\frac{\partial \overline{\rho}}{\partial x_j} + \frac{0.45q\overline{\rho'\rho'}}{\Lambda}
\tag{C.11}
$$

where the subscripts i, j, and k can take on the values of 1, 2 or 3. Hence, Equations C.9 represents six equations and Equation C.10 represents three equations, in general. A total of five model coefficients are contained in the above equations. These coefficients were determined from comparing model results with data from critical laboratory experiments where only one or two of the turbulent transport processes are dominant. These coefficients remained "invariant" in application of the equilibrium closure model and the Reynolds stress model (Sheng, 1982, 1984; Sheng and Villaret, 1989).

As shown in Sheng et al. (1990b), q^2 can be determined from the following dimensional equations when the mean flow variables are know

$$
\begin{aligned}
3A^2b^2sQ^4 \quad & + \quad A\left[(bs + 3b + 7b^2s)Ri - Abs(1 - 2b)\right]Q^2 \\
& + \quad b(s + 3 + 4bs)Ri^2 + (bs - A)(1 - 2b)Ri = 0
\end{aligned}
\tag{C.12}
$$

where A, b and s are model constants and

$$
Q = \frac{q}{\Lambda\sqrt{\left(\frac{\partial u}{\partial z}\right)^2 + \left(\frac{\partial v}{\partial z}\right)^2}}
\tag{C.13}
$$

$$
Ri = -\frac{\frac{g}{R_o}\frac{\partial \rho}{\partial z}}{\left(\frac{\partial u}{\partial z}\right)^2 + \left(\frac{\partial v}{\partial z}\right)^2}
\tag{C.14}
$$

where q is the total root mean square turbulent velocity and Λ is the turbulence macroscale.

One the mean flow variables are determined at each time step, the total root mean squared velocity, q, can be obtained from the above equations. After which, the vertical eddy viscosity and diffusivity can be calculated from

$$A_v = S_m \Lambda q \tag{C.15}$$

$$K_v = S_\rho \Lambda q \tag{C.16}$$

where S_m and S_ρ are defined as,

$$S_m = \frac{A + \overline{\omega}}{A - \omega} \frac{\overline{ww}}{q^2} \tag{C.17}$$

$$S_\rho = \frac{bs}{(bs - \omega)A} \frac{\overline{ww}}{q^2} \tag{C.18}$$

where $\omega = \frac{Ri}{AQ^2}$, $\overline{\omega} = \frac{\omega}{1 - \frac{\omega}{bs}}$ and

$$\overline{ww} = \frac{1 - 2b}{3(1 - 2\overline{\omega})} q^2 \tag{C.19}$$

Model constants in the above equations are $b = 0.125$, $s = 1.8$ and $A = 0.75$.

Λ, the turbulence macroscale, is subject to the following constraints

$$\left| \frac{d\Lambda}{dz} \right| \leq 0.65 \tag{C.20}$$

$$\Lambda \leq \frac{q}{N} = \frac{q}{\sqrt{-\frac{g}{\rho} \frac{\partial \rho}{\partial z}}} \tag{C.21}$$

$$\Lambda \leq C_1 \cdot H \tag{C.22}$$

$$\Lambda \leq C_1 \cdot H_p \tag{C.23}$$

$$\Lambda \leq C_2 \cdot \left(z_{q = q_{max}} - z_{q_{max/2}} \right) \tag{C.24}$$

where C_1 is usually between 0.1 and 0.25, H is the total depth, H_p is the depth of the pycnocline, C_2, which is between 0.1 and 0.25, is the fractional cut-off limitation of the turbulent macroscale and N is the Brunt-Väisälä frequency.

APPENDIX D
CH3D INTERNAL MODE

After solving the shallow water wave equations for the surface elevation, ζ and the vertically integrated velocities, U and V, the next step is to solve for the three-dimensional deficit velocity components, $\tilde{u}$ and $\tilde{v}$. These are defined by subtracting the vertical integrated velocities from the three-dimensional velocities,

$$
\begin{aligned}
\tilde{u} &= u - \frac{U}{H} \\
\tilde{v} &= v - \frac{V}{H}
\end{aligned}
\tag{D.1}
$$

The equations of motion for the deficit velocities are obtained by taking the three-dimensional equations (Equations 2.12 and 2.13) and subtracting the vertically integrated equations (Equations 2.31 and 2.32) resulting in the following equations for the deficit velocities

$$
\begin{aligned}
\frac{\partial \tilde{u} H}{\partial t} - \frac{E_v}{H} \frac{\partial}{\partial \sigma} \left(A_v \frac{\partial u}{\partial \sigma} \right) = \; & + \; \frac{g_{12}}{\sqrt{g_o}} H u - \frac{g_{12}}{\sqrt{g_o}} U + \frac{g_{22}}{\sqrt{g_o}} H v - \frac{g_{22}}{\sqrt{g_o}} V \\
& + \; \frac{R_o}{g_o} \Big[(\xi - \text{Dir. Inertia Terms}) \\
& \qquad - \int_{-1}^{0} (\xi - \text{Dir. Inertia Terms}) d\sigma \Big] \\
& - \; \tau_{s\xi} + \tau_{b\xi} \\
& - \; \frac{H R_o}{F^2} \Big[(\xi - \text{Dir. Baroclinic Terms}) \\
& \qquad - \int_{-1}^{0} (\xi - \text{Dir. Baroclinic Terms}) d\sigma \Big] \\
& + \; H E_H A_H \Big[(\xi - \text{Dir. Horizontal Diffusion Terms}) \\
& \qquad - \int_{-1}^{0} (\xi - \text{Dir. Horizontal Diffusion Terms}) d\sigma \Big]
\end{aligned}
\tag{D.2}
$$

$$\frac{\partial \tilde{v} H}{\partial t} - \frac{E_v}{H}\frac{\partial}{\partial \sigma}\left(A_v \frac{\partial v}{\partial \sigma}\right) = \begin{aligned}&- & \frac{g_{11}}{\sqrt{g_o}}Hu + \frac{g_{11}}{\sqrt{g_o}}U - \frac{g_{21}}{\sqrt{g_o}}Hv + \frac{g_{21}}{\sqrt{g_o}}V\\ &- & \frac{R_o}{g_o}\Big[(\eta-\text{Dir. Inertia Terms})\\ & & - \int_{-1}^{0}(\eta-\text{Dir. Inertia Terms})d\sigma\Big]\\ &- & \tau_{s\eta} + \tau_{b\eta}\\ &- & \frac{HR_o}{F_r^2}\Big[(\eta-\text{Dir. Baroclinic Terms})\\ & & - \int_{-1}^{0}(\eta-\text{Dir. Baroclinic Terms})d\sigma\Big]\\ &+ & HE_H A_H\Big[(\eta-\text{Dir. Horizontal Diffusion Terms})\\ & & - \int_{-1}^{0}(\eta-\text{Dir. Horizontal Diffusion Terms})d\sigma\Big] \end{aligned}$$

$$(D.3)$$

The terms on the right hand side of the equations are lumped into the single terms M_{xi} and M_η. Incorporating the definitions of the deficit velocities and the lumped together terms, Equations D.2 and D.3 can be rewritten as

$$\frac{\partial \tilde{u} H}{\partial t} - \frac{E_v}{H}\frac{\partial}{\partial \sigma}\left(A_v \frac{\partial}{\partial \sigma}\left(\tilde{u} + \frac{U}{H}\right)\right) = M_\xi \qquad (D.4)$$

$$\frac{\partial \tilde{v} H}{\partial t} - \frac{E_v}{H}\frac{\partial}{\partial \sigma}\left(A_v \frac{\partial}{\partial \sigma}\left(\tilde{v} + \frac{V}{H}\right)\right) = M_\eta \qquad (D.5)$$

The numerical solution of these equations treats the time derivatives and the vertical diffusion terms implicitly to avoid instability. The vertically integrated velocities have already been solved for at the $n+1$ time level and are considered knowns in the equations and are added to the M terms creating $\hat{M}$ terms. The lumped terms, $\hat{M}_\xi$ and $\hat{M}_\eta$, are taken at the n time level. Applying a forward time difference on the time derivative and central space derivative on the diffusion terms gives

$$\begin{aligned}\tilde{u}_{i,j,k}^{n+1} &- \frac{E_v}{(H_{i,j}^{n+1})^2}\frac{\Delta t}{\Delta \sigma_{i,j,k}}\left(A_v \frac{(\tilde{u}_{i,j,k+1}^{n+1} - \tilde{u}_{i,j,k}^{n+1})}{\Delta \sigma_{i,j,k+\frac{1}{2}}} - A_v \frac{(\tilde{u}_{i,j,k}^{n+1} - \tilde{u}_{i,j,k-1}^{n+1})}{\Delta \sigma_{i,j,k-\frac{1}{2}}}\right)\\ &= \frac{\tilde{u}_{i,j,k}^{n+1}H_{i,j}^{n+1}}{H_{i,j}^{n+1}} + \frac{\Delta t}{H_{i,j,k}^{n+1}}(\hat{M}_\xi)_{i,j,k}^n \end{aligned}$$

$$(D.6)$$

$$\begin{aligned}\tilde{v}_{i,j,k}^{n+1} &- \frac{E_v}{(H_{i,j}^{n+1})^2}\frac{\Delta t}{\Delta \sigma_{i,j,k}}\left(A_v \frac{(\tilde{v}_{i,j,k+1}^{n+1} - \tilde{v}_{i,j,k}^{n+1})}{\Delta \sigma_{i,j,k+\frac{1}{2}}} - A_v \frac{(\tilde{v}_{i,j,k}^{n+1} - \tilde{v}_{i,j,k-1}^{n+1})}{\Delta \sigma_{i,j,k-\frac{1}{2}}}\right)\\ &= \frac{\tilde{v}_{i,j,k}^{n+1}H_{i,j}^{n+1}}{H_{i,j}^{n+1}} + \frac{\Delta t}{H_{i,j,k}^{n+1}}(\hat{M}_\eta)_{i,j,k}^n \end{aligned}$$

$$(D.7)$$

The solution of these equations is performed over the vertical within a single horizontal cell resulting in a computationally efficient tridiagonal system of equations.

The calculation of the vertical velocities is performed using the three-dimensional continuity equation (Equation 2.11). A forward time finite difference is applied to the surface slope term and a centered space on the velocity gradients yielding

$$
\omega_{i,j,k}^{n+1} = \omega_{i,j,k-1}^{n+1} \; - \; \frac{\Delta\sigma}{\beta H_{i,j}^{n+1}} \frac{\zeta_{i,j}^{n+1} - \zeta_{i,j}^{n}}{\Delta t}
$$
$$
- \; \frac{1}{\sqrt{g_0} H_{i,j}^{n+1}} \left[\frac{(\sqrt{g_0} Hu)_{i+1,j,k}^{n+1} - (\sqrt{g_0} Hu)_{i,j,k}^{n+1}}{\Delta\xi} + \frac{(\sqrt{g_0} Hv)_{i,j+1,k}^{n+1} - (\sqrt{g_0} Hv)_{i,j,k}^{n+1}}{\Delta\eta} \right]
$$
$$
\tag{D.8}
$$

In this equation, all the u and v terms at the $n+1$ time level are known from previous solutions. The $\omega_{i,j,k-1}^{n+1}$ value in the bottom cell is equal to zero from the boundary conditions, therefore a solution for each of the vertical indices can be obtained by stepping the solution from the bottom cell to the surface, in each case the vertical velocities for the next cell up have been calculated from the solution of the cell below.

APPENDIX E
CH3D SALINITY FDE

The finite difference solution of the advection-diffusion equation for salinity follow closely the method used for the internal mode solution. The time derivatives and the vertical diffusion terms are treated implicitly with the horizontal diffusion and advection terms treated explicitly. Applying the finite difference formulation to Equation 2.21 yields

$$
H_{i,j}^{n+1} S_{i,j}^{n+1} - \frac{\Delta t E_v}{H_{i,j}^{n+1} S_{c_v}} \frac{1}{\Delta \sigma_{i,j,k}} \left[\frac{D_v}{\Delta \sigma_{i,j,k+\frac{1}{2}}} (S_{i,j,k+1}^{n+1} - S_{i,j,k}^{n+1}) - \frac{D_v}{\Delta \sigma_{i,j,k-\frac{1}{2}}} (S_{i,j,k}^{n+1} - S_{i,j,k-1}^{n+1}) \right]
$$
$$
= H_{i,j}^n S_{i,j,k}^n + \Delta t N_{i,j,k}^n
$$
(E.1)

where $N_{i,j,k}^n$ are the advection and horizontal diffusion terms treated at the n time level. This formulation leads to a tridiagonal matrix solution as in the internal mode.

One of the most basic transport schemes is the upwind method. This method takes advantage of the concept that the velocity moving out of a cell will tend to transport concentrations equal to the cell concentrations. From Equation 2.21 the finite difference form of the advection terms can be written as

$$
\frac{\partial}{\partial \xi} \left(\sqrt{g_o} H u S \right) = \frac{1}{\Delta \xi} \left[\begin{array}{l} \left[\begin{array}{l} \frac{1}{2}(u_{i+1,j,k} - |u_{i+1,j,k}|)S_{i+1,j,k} \\ +\frac{1}{2}(u_{i+1,j,k} + |u_{i+1,j,k}|)S_{i,j,k} \end{array} \right] (\sqrt{g_o})_{i+1,j,u} H_{i+1,j,u} \\ - \left[\begin{array}{l} \frac{1}{2}(u_{i,j,k} - |u_{i,j,k}|)S_{i,j,k} \\ +\frac{1}{2}(u_{i,j,k} + |u_{i,j,k}|)S_{i-1,j,k} \end{array} \right] (\sqrt{g_o})_{i,j,u} H_{i,j,u} \end{array} \right]
$$
(E.2)

The horizontal diffusion second derivative terms can be split into four separate terms and can be written in finite difference form as

$$
\frac{E_h}{S_{c_H} \sqrt{g_o}} \left[\frac{\partial}{\partial \xi} \left(\sqrt{g_o} H g^{11} \frac{\partial S}{\partial \xi} \right) \right] = \frac{(\sqrt{g_o} H g^{11})_{i+\frac{1}{2},j} \frac{S_{i+1,j,k} - S_{i,j,k}}{\Delta \xi} - (\sqrt{g_o} H g^{11})_{i-\frac{1}{2},j} \frac{S_{i,j,k} - S_{i-1,j,k}}{\Delta \xi}}{\Delta \xi}
$$
(E.3)

226

$$\frac{E_h}{S_{c_H}\sqrt{g_o}}\left[\frac{\partial}{\partial\xi}\left(\sqrt{g_o}Hg^{12}\frac{\partial S}{\partial\eta}\right)\right] = \frac{\left(\sqrt{g_o}Hg^{12}\right)_{i+\frac{1}{2},j}\frac{S_{UR}-S_{LR}}{\Delta\eta}-\left(\sqrt{g_o}Hg^{12}\right)_{i-\frac{1}{2},j}\frac{S_{UL}-S_{LL}}{\Delta\eta}}{\Delta\xi} \tag{E.4}$$

$$\frac{E_h}{S_{c_H}\sqrt{g_o}}\left[\frac{\partial}{\partial\eta}\left(\sqrt{g_o}Hg^{21}\frac{\partial S}{\partial\xi}\right)\right] = \frac{\left(\sqrt{g_o}Hg^{21}\right)_{i,j+\frac{1}{2}}\frac{S_{UR}-S_{UL}}{\Delta\xi}-\left(\sqrt{g_o}Hg^{21}\right)_{i,j-\frac{1}{2}}\frac{S_{LR}-S_{LL}}{\Delta\xi}}{\Delta\eta} \tag{E.5}$$

$$\frac{E_h}{S_{c_H}\sqrt{g_o}}\left[\frac{\partial}{\partial\eta}\left(\sqrt{g_o}Hg^{22}\frac{\partial S}{\partial\eta}\right)\right] = \frac{\left(\sqrt{g_o}Hg^{22}\right)_{i,j+\frac{1}{2}}\frac{S_{i,j+1,k}-S_{i,j,k}}{\Delta\eta}-\left(\sqrt{g_o}Hg^{22}\right)_{i,j-\frac{1}{2}}\frac{S_{i,j,k}-S_{i,j-1,k}}{\Delta\eta}}{\Delta\eta}$$
$$\tag{E.6}$$

where the subscripts UR, LR, UL, and LL stand for upper right, lower right, upper left and lower left corners, respectively and are defined as

$$S_{UR} = \frac{1}{4}\left(S_{i,j,k} + S_{i+1,j,k} + S_{i,j+1,k} + S_{i+1,j+1,k}\right) \tag{E.7}$$

$$S_{LR} = \frac{1}{4}\left(S_{i,j,k} + S_{i+1,j,k} + S_{i,j-1,k} + S_{i+1,j-1,k}\right) \tag{E.8}$$

$$S_{UL} = \frac{1}{4}\left(S_{i-1,j,k} + S_{i,j,k} + S_{i-1,j+1,l} + S_{i,j+1,k}\right) \tag{E.9}$$

$$S_{LL} = \frac{1}{4}\left(S_{i-1,j,k} + S_{i,j,k} + S_{i-1,j-1,l} + S_{i,j-1,k}\right) \tag{E.10}$$

APPENDIX F
PARALLEL SOLUTION OF TRIDIAGONAL SYSTEMS

As was shown in Chapter 2, tridiagonal systems of equations can result during the process of numerical modeling. Since a major portion of modeling time will be spent solving these systems, it is important that an efficient algorithm is used. The choice of algorithms is further complicated by the presence of many parallel algorithms as well. This appendix compares several serial and parallel algorithms known for solving a tridiagonal system of linear equations. The techniques considered are Gaussian elimination, Gaussian elimination with partial pivoting, cyclic reduction and conjugate gradient. Rather than applying these algorithms to full CH3D code, a simple 1D code was written to model flow in the Delaware Estuary. This 1D code has only 1 tridiagonal system to solve as opposed to the 4 tridiagonal systems (Equations 2.40, 2.45, 2.52, and 2.57) of the CH3D code, thus providing for simpler analysis.

F.1 Computing Platform

The computing environment chosen to run the algorithms was a Silicon Graphics (SGI) Origin 2000. Details about this symmetric multi-processor machine can be found in Table F.1. No disk or network access occurred during the timing of runs; therefore, those relevant hardware devices are not listed. The algorithms were coded in Fortran-77 (using some extensions) and compiled with the optimization options '-n32 -mips4 -O3'. For the Power Fortran Accelerator (PFA) code, the option '-pfa' was used. PFA is a Fortran preprocessor that discovers parallelism in any sequential code and converts it to an equivalent parallel code. All runs were executed exclusively on the machine such that the processors could be devoted entirely to the given code.

Table F.1: Specifications for the SGI Origin 2000.

FPU: MIPS R10010 Floating Point Chip Revision: 0.0
CPU: MIPS R10000 Processor Chip Revision: 2.6
4 195 MHZ IP27 Processors
Main memory size: 512 Megabytes
Instruction cache size: 32 Kilobytes
Data cache size: 32 Kilobytes
Secondary unified instruction/data cache size: 4 Megabytes

Separate codes were created for each matrix size in order to prevent problems arising from over-dimensioning arrays.

Two versions of parallel codes were tested. One was the parallel code that we came up with manually and the other was the code generated by PFA. PFA is a product of Kuck & Associates, Inc. and it is a part of the standard software package supplied by SGI. Although a considerable number of command line options, directives and assertions are available none were used.

F.2   Description of Algorithms

Four of the most popular algorithms for solving a tridiagonal system have been chosen for comparison. These algorithms are Gaussian elimination, Gaussian elimination with partial pivoting, cyclic reduction and conjugate gradient.

Unfortunately, the cyclic reduction algorithm has restrictions on input size. This leads to the algorithm performing more work than necessary to solve an arbitrary size matrix, thus all matrices used in this paper will conform to an optimal cyclic reduction input.

Letting $M = N - 1$ and $N = 2^n$, a linear tridiagonal system of equations can be represented as

$$\mathbf{A}x = \mathbf{d} \tag{F.1}$$

where

$$\mathbf{A} = \begin{bmatrix} b_1 & c_1 & & & & \\ a_2 & b_2 & c_2 & & & \\ & a_3 & b_3 & c_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & a_{M-1} & b_{M-1} & c_{M-1} \\ & & & & a_M & b_M \end{bmatrix} \qquad (F.2)$$

$$\mathbf{x} = (x_1, x_2, \ldots, x_M)^T \qquad (F.3)$$

and

$$\mathbf{d} = (d_1, d_2, \ldots, d_M)^T \qquad (F.4)$$

Alternatively, Equation F.1 can be written as

$$a_i x_{i-1} + b_i x_i + c_i x_{i+1} = d_i \quad \text{for} \quad i = 1, 2, \ldots, M \qquad (F.5)$$

where

$$x_0 = a_1 = 0 \quad \text{and} \quad x_{M+1} = c_M = 0 \qquad (F.6)$$

For the purposes of this paper, $N = 5, 6, \ldots, 15$, resulting in the following values of M: 31, 63, 127, 255, 511, 1023, 2047, 4095, 8191, 16383, and 32767.

F.2.1 Gaussian Elimination

Gaussian elimination is one of the oldest direct methods for solving a system of equations. The diagonal elements are used to eliminate the entries in the columns below them. For tridiagonal matrices a simple recursive method is presented below (Carnahan et al., 1969).

Setting $x_M = \gamma_M$, the remaining x_i's can be solved for recursively, $i = N - 1, N - 2, \ldots, 1$, using

$$x_i = \gamma_i - \frac{c_i x_{i+1}}{\beta_i} \qquad (F.7)$$

where $\beta_1 = b_1$ and $\gamma_1 = \frac{d_1}{\beta_1}$, the remaining β_i's and γ_i's can be determined recursively, $i = 2, 3, \ldots, N$, using

$$\beta_i = b_i - \frac{a_i c_{i-1}}{\beta_{i-1}} \qquad (F.8)$$

$$\gamma_i = \frac{d_i - a_i \gamma_{i-1}}{\beta_i} \tag{F.9}$$

F.2.2 Gaussian Elimination with Partial Pivoting

While Gaussian elimination is a simple algorithm it is prone to several problems. If very small numbers appear in the diagonal positions, numerical problems can develop. Therefore, as the algorithm proceeds, it scans down through the remaining diagonals and swaps the row with the largest diagonal value. This swapping technique is called partial pivoting.

Instead of writing a new program, we employed the LAPACK routine SGTSV. LAPACK is a library of Fortran 77 routines for solving common linear algebra problems (Anderson et al., 1990). All of the specifics of the LAPACK routines can be found in the User's Guide (Anderson et al., 1995). In addition, performance of various LAPACK routines are described in Anderson and Dongarra (1993).

F.2.3 Cyclic Reduction

The cyclic reduction algorithm, though more complicated than the Gaussian elimination algorithms, is highly parallelizable (Lakshmivarahan and Dhall, 1990). The algorithm splits the matrix in half and recursively eliminates variables. However, as mentioned earlier, the cyclic reduction method requires the input matrix to be of a specific size. Arbitrary size matrices can be padded with zeros to convert them to the appropriate size. The algorithm consists of two phases, the reduction phase and the back substitution phase.

Reduction Phase

The reduction phase for the cyclic reduction algorithm can be given as:

FOR $j = 1$ to $n - 1$; DO

    FOR $i \in \{2^j, 2 \times 2^j, 3 \times 2^j, \ldots, 2^n - 2^j\}$; DO IN PARALLEL

        COMPUTE $p_i^{(j)}$

    END

END

where

$$p_i^{(j)} = (a_i^{(j)}, b_i^{(j)}, c_i^{(j)}, d_i^{(j)}) \tag{F.10}$$

subject to

$$p_i^{(j)} = (0, 1, 0, 0), \quad \text{for} \quad i \leq 0 \quad \text{and} \quad i \geq N \tag{F.11}$$

and

$$x_i = 0, \quad \text{for} \quad i \leq 0 \quad \text{and} \quad i \geq N \tag{F.12}$$

Finally, letting $h = 2^{j-1}$, $(a_i^{(j)}, b_i^{(j)}, c_i^{(j)}, d_i^{(j)})$ can be solved for using

$$
\begin{aligned}
e_i^{(j)} &= -\frac{a_i^{(j-1)}}{b_{i-h}^{(j-1)}} \\
f_i^{(j)} &= -\frac{c_i^{(j-1)}}{b_{i+h}^{(j-1)}} \\
a_i^{(j)} &= e_i^{(j)} a_{i-h}^{(j-1)} \\
c_i^{(j)} &= f_i^{(j)} c_{i+h}^{(j-1)} \\
b_i^{(j)} &= b_i^{(j-1)} + e_i^{(j)} c_{i-h}^{(j-1)} + f_i^{(j)} a_{i+h}^{(j-1)} \\
d_i^{(j)} &= d_i^{(j-1)} + e_i^{(j)} d_{i-h}^{(j-1)} + f_i^{(j)} d_{i+h}^{(j-1)}
\end{aligned}
\tag{F.13}
$$

<u>Back Substitution Phase</u>

The first step of the back substitution phase is to solve for $x_{2^{n-1}}$ using

$$x_{2^{n-1}} = \frac{d_{2^{n-1}}^{(n-1)}}{b_{2^{n-1}}^{(n-1)}} \tag{F.14}$$

The back substitution phase for the cyclic reduction algorithm can be described as:

FOR $k = (n-1)$ to 1 STEP -1; DO

    FOR $i \in \{2^{k-1}, 3 \times 2^{k-1}, 5 \times 2^{k-1}, \ldots, 2^n - 2^{k-1}\}$; DO IN PARALLEL

    $h = 2^{k-1}$

        COMPUTE $x_i = \frac{d_i^{(k-1)} - a_i^{(k-1)} x_{i-h} - c_i^{(k-1)} x_{i+h}}{b_i^{(k-1)}}$

END

END

F.2.4 Conjugate Gradient

The conjugate gradient algorithm is an iterative method for solving a system of equations. A suitable form of the conjugate gradient method can be found below (Casulli and Cheng, 1992).

If we substitute $e_i = \sqrt{b_i} x_i$ into the alternative form of the tridiagonal system and divide by $\sqrt{b_i}$, Equation F.5 becomes

$$\frac{a_i}{\sqrt{b_i b_{i-1}}} e_{i-1} + e_i + \frac{c_i}{\sqrt{b_i b_{i+1}}} e_{i+1} = \frac{d_i}{\sqrt{b_i}} \quad \text{for} \quad i = 1, 2, \ldots, M \qquad (F.15)$$

Now, letting $\hat{a} = \frac{a_i}{\sqrt{b_i b_{i-1}}}$, $\hat{c} = \frac{c_i}{\sqrt{b_i b_{i+1}}}$, and $\hat{d} = \frac{d_i}{\sqrt{b_i}}$ the tridiagonal matrix can be represented as

$$e_i + \hat{a}_i e_{i-1} + \hat{c}_i e_{i+1} = \hat{d}_i \quad \text{for} \quad i = 1, 2, \ldots, M \qquad (F.16)$$

where e is the independent variable and $\hat{a}_i$, $\hat{c}_i$, and $\hat{d}_i$ are constant value coefficients. The conjugate gradient algorithm then proceeds as follows:

Step 1. Guess $e_i^{(0)}$

Step 2. Set $p_i^{(0)} = r_i^{(0)} = e_i^{(0)} + \hat{a}_i e_{i-1}^{(0)} + \hat{c}_i e_{i+1}^{(0)} - \hat{d}_i$

Step 3. Then for $k = 0, 1, 2, \ldots$ and until $(\mathbf{r}^{(k)}, \mathbf{r}^{(k)}) < \epsilon$, calculate

$$e_i^{(k+1)} = e_i^{(k)} - \alpha^{(k)} p_i^{(k)}, \qquad \text{where} \quad \alpha^{(k)} = \frac{(\mathbf{r}^{(k)}, \mathbf{r}^{(k)})}{(\mathbf{p}^{(k)}, Mp^{(k)})} \qquad (F.17)$$

$$r_i^{(k+1)} = r_i^{(k)} - \alpha^{(k)} (Mp^{(k)})_i \qquad (F.18)$$

$$p_i^{(k+1)} = r_i^{(k+1)} - \beta^{(k)} p_i^{(k)}, \qquad \text{where} \quad \beta^{(k)} = \frac{(\mathbf{r}^{(k+1)}, \mathbf{r}^{(k+1)})}{(\mathbf{r}^{(k)}, \mathbf{r}^{(k)})} \qquad (F.19)$$

In Equations F.17 and F.18, Mp is defined as

$$(Mp^{(k)})_i = p_i^{(k)} + \hat{a}_i p_{i-1}^{(k)} + \hat{c}_i p_{i+1}^{(k)} \tag{F.20}$$

F.3   Data Sets

F.3.1   Identity Matrices

Tridiagonal identity matrices serve as a simple test for the algorithms. Setting $a_2, a_3, \ldots, a_M = c_1, c_2, c_{M-1} = 0$, and letting $\tilde{d}_i = \frac{d_i}{b_i}$, Equation F.1 becomes

$$\mathbf{Ix} = \tilde{\mathbf{d}} \tag{F.21}$$

which has the solution

$$\mathbf{x} = (\frac{d_1}{b_1}, \frac{d_2}{b_2}, \ldots, \frac{d_M}{b_M})^T \tag{F.22}$$

Random numbers in the range $0, 1, \ldots, M-1$ were used for $\tilde{d}$ with one set for each matrix size.

F.3.2   1-D Hydrodynamic Model of the Delaware Estuary

The Delaware Estuary is a funnel-shaped estuary on the Atlantic coast of the United States (Figure F.1). It extends from Capes May (New Jersey) and Henlopen (Delaware) to the fall line in Trenton, New Jersey. The estuary measures 134 miles (216 km) along its axis and is approximately 1000 feet (305 m) wide at Trenton and 27 miles (44 km) wide at the widest point of Delaware Bay.

The estuary has the unique quality that both the width and cross sectional area can be represented well by exponential functions of distance along the axis of the estuary (Harleman, 1966). These features, along with its nearly constant depth, make the Delaware Estuary well suited to a one-dimensional numerical model. The exponential function are

$$\text{Width} = B = 1,000e^{\delta x} \tag{F.23}$$

$$\text{Cross Sectional Area} = A = 21,000e^{\delta x} \tag{F.24}$$

where $x = 0$ at Trenton, $\delta = 0.67 \times 10^{-5}$ and the mean depth $\bar{h} = \frac{A}{B} = 21$ feet.
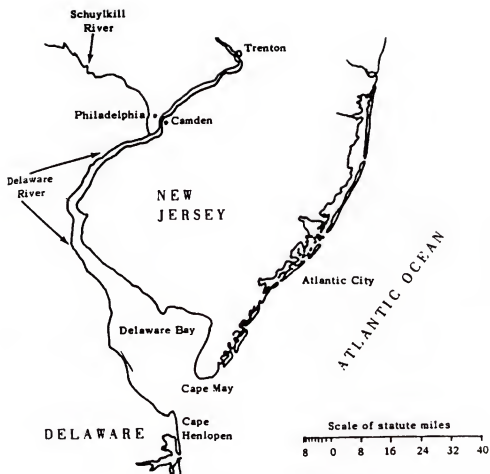
Figure F.1: The Delaware Bay Estuary (Harleman, 1966).

<u>Governing Equations</u>

The one dimensional, cross-sectionally averaged, vertically integrated continuity equation is

$$\frac{\partial \zeta}{\partial t} + \frac{1}{B}\frac{\partial (UB)}{\partial x} = 0 \tag{F.25}$$

The momentum equation, keeping only the propagation term, is

$$\frac{\partial U}{\partial t} + gH\frac{\partial \zeta}{\partial x} = 0 \tag{F.26}$$

The continuity and momentum equations are then written in an implicit finite difference form as

$$\frac{\zeta_i^{n+1} - \zeta_i^n}{\Delta t} + \frac{1}{B_{(s)i}}\frac{B_{(u)i+1}U_{i+1}^{n+1} - B_{(u)i}U_i^{n+1}}{\Delta x} = 0 \tag{F.27}$$

$$\frac{U_i^{n+1} - U_i^n}{\Delta t} + gH_{(u)i}^n\frac{\zeta_i^{n+1} - \zeta_{i-1}^{n+1}}{\Delta x} = 0 \tag{F.28}$$

The momentum FDE, Equation F.28, is solved for U_i^{n+1} and then inserted into the continuity FDE, Equation F.27 resulting in

$$a_i\zeta_{i-1}^{n+1} + b_i\zeta_i^{n+1} + c_i\zeta_{i+1}^{n+1} = d_i \tag{F.29}$$

where

$$a_i = -\left(\frac{\Delta t}{\Delta x}\right)^2 \frac{B_{(u)i}}{B_{(s)i}}gH_{(u)i}^n \tag{F.30}$$

$$c_i = -\left(\frac{\Delta t}{\Delta x}\right)^2 \frac{B_{(u)i+1}}{B_{(s)i}}gH_{(u)i+1}^n \tag{F.31}$$

$$b_i = 1 - a_i - c_i \tag{F.32}$$

$$d_i = \zeta_i^n - \frac{\Delta t}{\Delta x}\frac{B_{(u)i+1}U_{i+1}^n - B_{(u)i}U_i^n}{B_{(s)i}} \tag{F.33}$$

Equation F.29 represents a tridiagonal system of equations which changes for each time step in the model. The system of equations is not symmetric due to the changing width of the estuary.

F.4   Results

In this section we summarize our experimental results. The times listed are the average times taken to solve the given system of equations only. To strike a balance between the accuracy and the total time taken, each run had a minimum time of 10 seconds or was repeated a minimum of 10 times. The maximum number of iterations in the conjugate gradient algorithm was set to 10,000,000, although this value was never reached.

As mentioned earlier, the cyclic reduction algorithm also has a restriction on its input size. It must be $M = 2^n - 1$. Any system whose size is in the range $M = 2^n$ to $M = 2^{n+1} - 1$ must be padded with extra zeros up to $M = 2^{n+1}$. Thus, the CPU time for $M = 2^n$ would be twice that of $M = 2^n - 1$ even though only one additional equation is added.

F.4.1   Identity Systems

The cyclic reduction (PFA) algorithm and the conjugate gradient algorithms are not shown because they were always slower than the cyclic reduction and conjugate gradient (PFA), respectively.

Figure F.2 shows the CPU time taken to solve the the identity systems. The conjugate gradient method with 4 processors solves the systems the fastest. However, because the systems being solved are identity there is only one iteration of the conjugate gradient routine; thus, this time is somewhat deceiving. The serial version of the Gaussian elimination with partial pivoting is nearly as fast with only one processor. It is also noted that the serial Gaussian elimination with partial pivoting is faster the normal Gaussian elimination algorithm.

Figures F.3, F.4, and F.5 show the speedup attained using 2, 3, and 4 processors, respectively. All four methods reach approximately the same limit. The Gaussian elimination algorithm did not speed up well while the conjugate gradient algorithm reached the highest speedups.
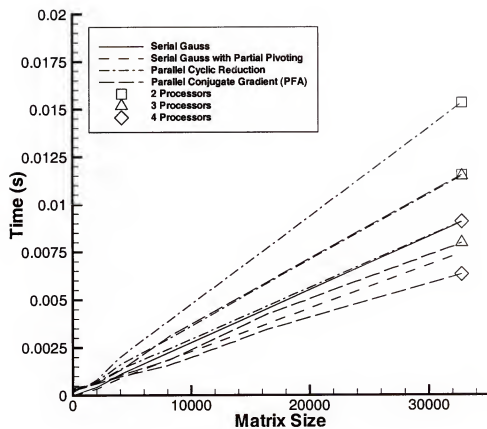
Figure F.2: Average CPU time taken to solve the identity systems.
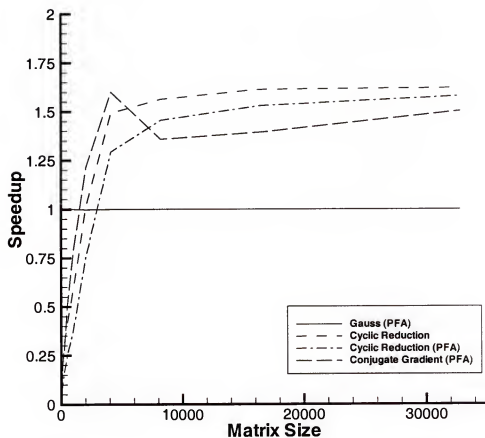
Figure F.3: Parallel speedup gained in solving the identity systems using 2 processors.
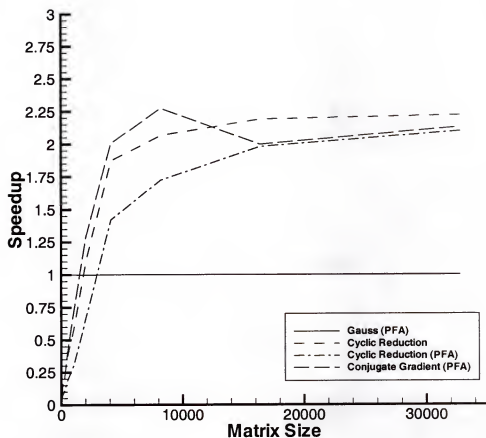
Figure F.4: Parallel speedup gained in solving the identity systems using 3 processors.
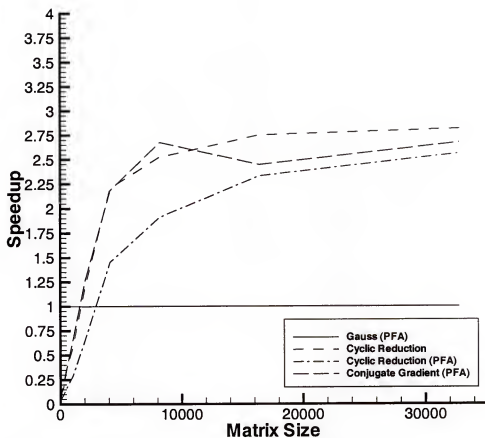
Figure F.5: Parallel speedup gained in solving the identity systems using 4 processors.

F.4.2   1-D Hydrodynamic Model of the Delaware Estuary

The hydrodynamic code modeled 5 days of real time. The first 4.5 days used the serial Gaussian elimination with the last half day being any one of the algorithms. The last half day's systems were the only ones timed. The simulation included an additional bottom friction term to maintain model stability.

Again, the cyclic reduction (PFA) algorithm and the conjugate gradient algorithms are not shown because they were always slower than the cyclic reduction and conjugate gradient (PFA), respectively.

Figure F.6 shows the average CPU times taken to solve the tridiagonal systems generated by the hydrodynamic model. The conjugate gradient algorithms are not shown because they take considerably longer. Figures F.7 show the average CPU times of the conjugate gradient algorithms with an error tolerance, $\epsilon = 1.0 \times 10^{-5}$. The CPU times decrease with an increased number of processors and they increase with increasing Δt. The increase in CPU time derives its origin from the algorithm being iterative in nature. With a longer amount of time between successive solving of the system, more iterations of the conjugate gradient are required.

The tolerance also plays an important role in the CPU time as is shown in Figure F.8. By changing the tolerance to a larger number, the systems are solved nearly twice as fast, although still slower than the direct methods. A tolerance of 1.0×10^{-5} produces results nearly identical to the direct methods, while a tolerance of 1.0×10^{-2} produces answers which differ from the direct methods by $O(0.1)$.

Figure F.9, F.10, and F.11 show the speedup attained using 2, 3, and 4 processors, respectively. The Gaussian elimination algorithm did not speed up well while the conjugate gradient algorithm reached the highest speedups. Comparing these plots to those of the identity systems, it can be seen that the cyclic reduction and conjugate gradient algorithms do not speed up as much.
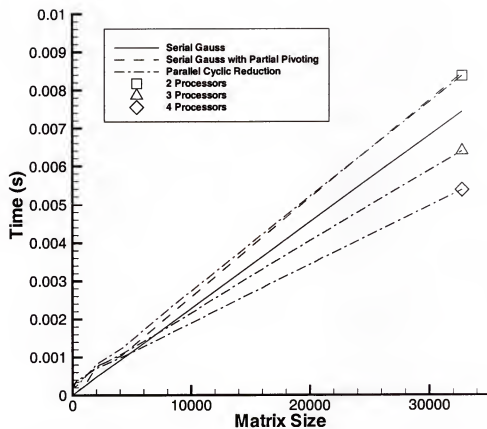
Figure F.6: Average CPU time taken to solve the tridiagonal systems generated by the hydrodynamic model.
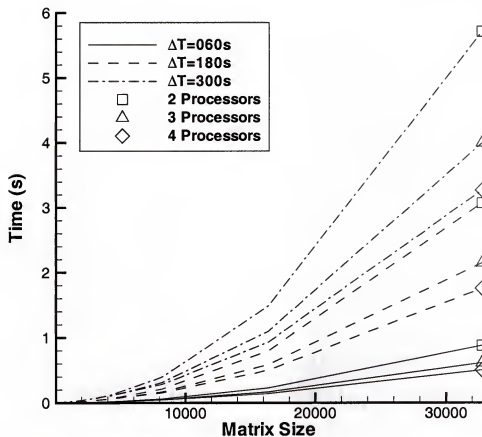
Figure F.7: Average CPU time taken to solve the tridiagonal systems generated by the hydrodynamic model using the conjugate gradient method with $\epsilon = 1.0 \times 10^{-5}$.
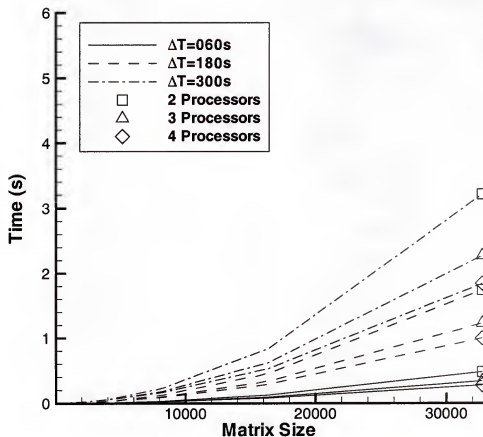
Figure F.8: Average CPU time taken to solve the tridiagonal systems generated by the hydrodynamic model using the conjugate gradient method with $\epsilon = 1.0 \times 10^{-2}$.
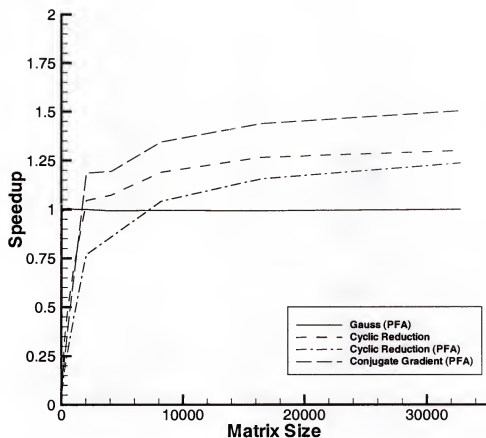
Figure F.9: Parallel speedup gained in solving the hydrodynamic model's tridiagonal systems using 2 processors. The conjugate gradient algorithm uses $\epsilon = 1.0 \times 10^{-5}$.
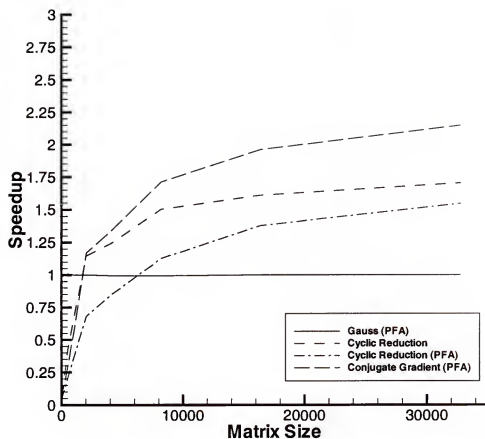
Figure F.10: Parallel speedup gained in solving the hydrodynamic model's tridiagonal systems using 3 processors. The conjugate gradient algorithm uses $\epsilon = 1.0 \times 10^{-5}$.
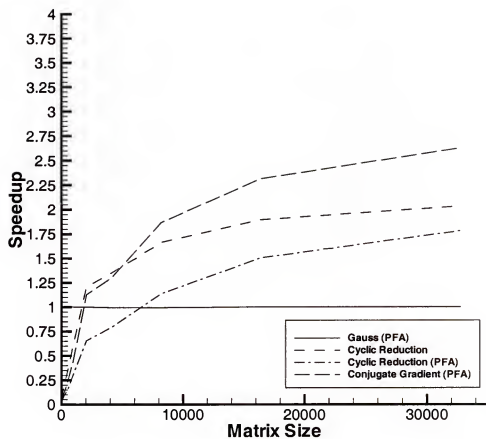
Figure F.11: Parallel speedup gained in solving the hydrodynamic model's tridiagonal systems using 4 processors. The conjugate gradient algorithm uses $\epsilon = 1.0 \times 10^{-5}$.

F.5   Conclusions

From the results shown, several important conclusions about using the algorithms for tridiagonal systems generated by a 1-D hydrodynamic model, can be drawn. Firstly, the multi-processor cyclic reduction algorithm is the fastest for solving tridiagonal systems. However, with 4 processors, the cyclic reduction algorithm is only approximately 40% faster implying that the cyclic reduction algorithm is performing more total work. Secondly, for the hydrodynamic tridiagonal systems, the Gaussian elimination without partial pivoting is faster than the algorithm with pivoting. Lastly, one may be able to get the conjugate gradient algorithm to compare well with the direct solvers if a small enough time step is used with the numerical model. However, by decreasing this value, the time necessary to run the numerical model for a fixed length of time becomes longer than the less restrictive direct methods.

More importantly, it is noted that a large number of simultaneous equations are required before a speedup of 1 is reached. Any time a speedup is less than 1, it would be more advantageous not to use the parallel algorithm. For the Gauss (PFA), cyclic reduction and cyclic reduction (PFA), the hydrodynamic model needs to have at least 2000 cells while for the conjugate gradient, at least 6000 cells are required.

Table F.2 shows the sweep lengths of two typical CH3D grid systems, the Florida Bay grids used in the wetting and drying study of Davis (1996). Approximately 37% of the cells are covered with water in both the coarse and fine grids (97×74 and 194×148 cells, respectively). The average size of the tridiagonal sweeps is 28 for the coarse grid and 43 cells for the fine grid. Thus, the small sweep lengths of these typical grid systems indicate that it would not be advantageous to use one of the aforementioned parallel algorithms. To obtain the best possible execution times, the serial gauss algorithm should be used.

Table F.2: Sweep lengths in the boundary-fitted Florida Bay coarse and fine grids (97×74 and 194×148 cells, respectively).

Grid	Sweep Direction	Total Number of Sweeps	Average Length of Sweeps
coarse	i	184	30
	j	234	24
	k	195	29
fine	i	500	45
	j	568	39
	k	504	45

APPENDIX G
CAUSEWAY LOCATIONS IN THE CH3D IRL FINE GRID

The following figures (Figures G.1 through G.4) illustrate the 16 causeways resolved by the CH3D fine grid within the IRL domain. Because of the long, thin nature of the causeways, the causeways are defined with a series of thin-wall barriers which block flow through one side of a given grid cell.
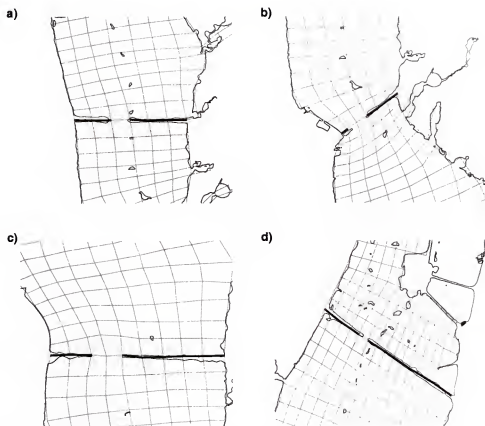
Figure G.1: Locations of the thin wall barriers (dark, thick lines) used to resolve the a) Train, b) Brewer, c) NASA West and d) NASA East Causeways. The fine grid is shown with dotted lines.
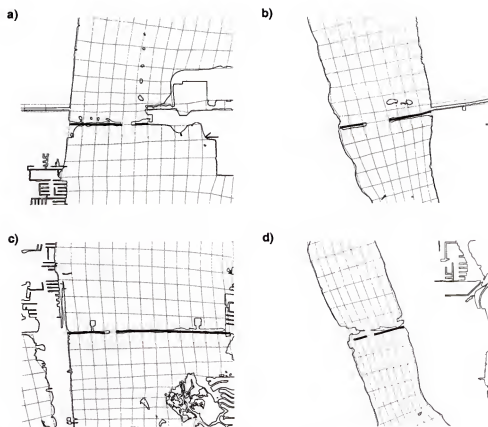
Figure G.2: Locations of the thin wall barriers (dark, thick lines) used to resolve the a) Bennett East, b) Bennett West, c) Merritt and d) Hubert Humphrey Causeways. The fine grid is shown with dotted lines.
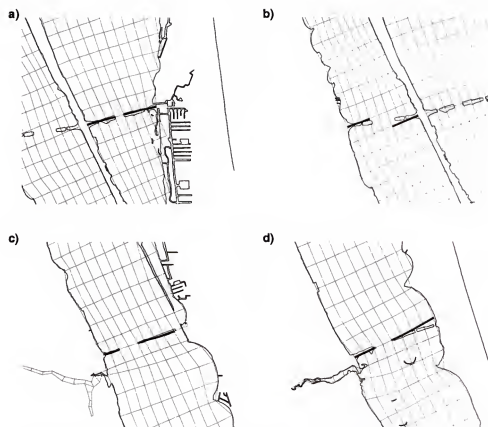
Figure G.3: Locations of the thin wall barriers (dark, thick lines) used to resolve the a) Pineda East, b) Pineda West, c) Eau Gallie and d) Melbourne Causeways. The fine grid is shown with dotted lines.
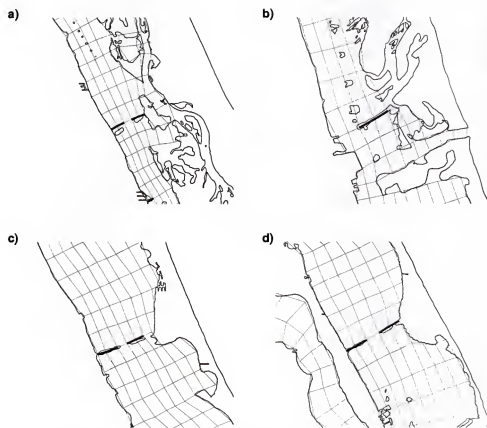
Figure G.4: Locations of the thin wall barriers (dark, thick lines) used to resolve the a) Wabasso, b) North Beach, c) 732 and d) A1A (at St. Lucie) Causeways. The fine grid is shown with dotted lines.

APPENDIX H
WATER LEVEL AND SALINITY STATION LOCATIONS IN THE CH3D IRL FINE GRID

The following figures illustrate the exact locations of the water level and salinity measurement stations within the IRL fine grid. The measuring stations shown in Figures H.1 through H.4 are located in Ponce Inlet (FDEP #872-1147), Mosquito Lagoon (FDEP #872-1164), Titusville (FDEP #872-1456), Merritt Causeway East (FDEP #872-1647), Merritt Causeway West (FDEP #872-1648), Banana River (FDEP #872-1789), Melbourne (FDEP #872-1843), Sebastian Inlet (FDEP #872-2004), Vero Bridge (FDEP #872-2125), Ft. Pierce Causeway (FDEP #872-2208), Ft. Pierce Inlet(FDEP #872-2213), St. Lucie Inlet Inlet (FDEP #872-2375), Haulover Canal (USGS #02248380), Sebastian River (USGS #275017080295600) and Wabasso (USGS #02251800).
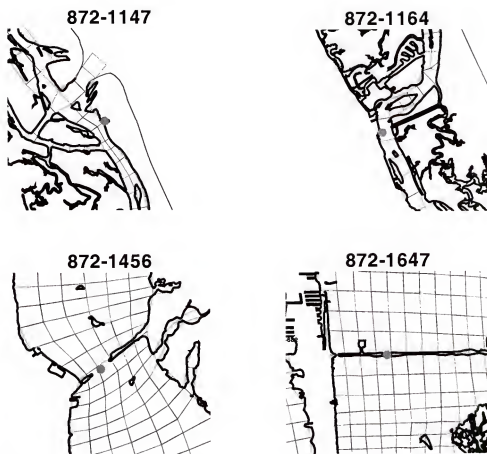
Figure H.1: Locations of the Ponce Inlet (FDEP #872-1147), Mosquito Lagoon (FDEP #872-1164), Titusville (FDEP #872-1456) and Merritt Causeway East (FDEP #872-1647) stations.
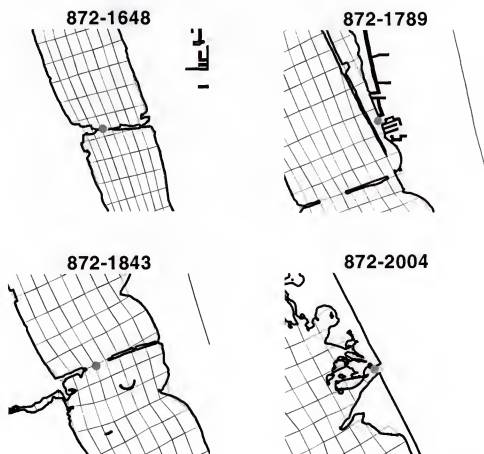
Figure H.2: Locations of the Merritt Causeway West (FDEP #872-1648), Banana River (FDEP #872-1789), Melbourne (FDEP #872-1843) and Sebastian Inlet (FDEP #872-2004) stations.

259
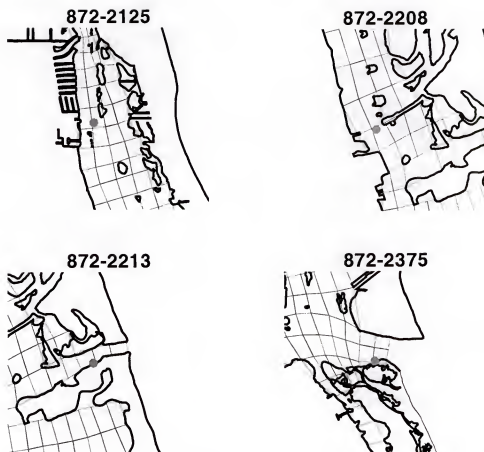
Figure H.3: Locations of the Vero Bridge (FDEP #872-2125), Ft. Pierce Causeway (FDEP #872-2208), Ft. Pierce Inlet(FDEP #872-2213) and St. Lucie Inlet Inlet (FDEP #872-2375) stations.
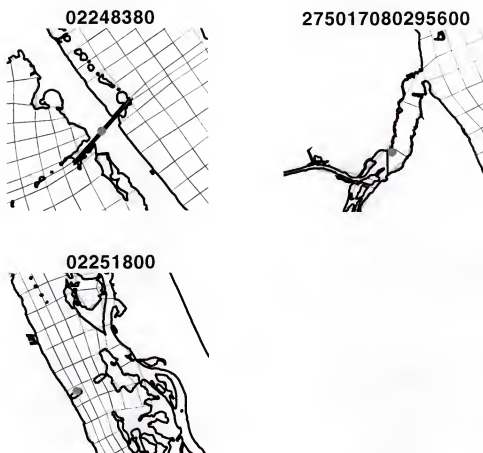
Figure H.4: Locations of the Haulover Canal (USGS #02248380), Sebastian River (USGS #275017080295600) and Wabasso (USGS #02251800) stations.

APPENDIX I
PRECIPITATION AND EVAPORATION PLOTS

The following figures show the monthly and yearly totals of precipitation, evaporation and combined precipitation and evaporation over the Indian River Lagoon Domain. Figure I.1 shows the precipitation into the lagoon as measured by the 13 SJRWMD stations given in Table 4.1. Figures I.5 through I.7 show the evaporation and combined precipitation and evaporation using the percentile formulation for evaporation at Vero Beach. Figures I.8 through I.10 show the evaporation and combined precipitation and evaporation using the spatial percentile formulation. Figures I.11 through I.13 show the evaporation and combined precipitation and evaporation using the evaporation data measured at the Belle Glade station.
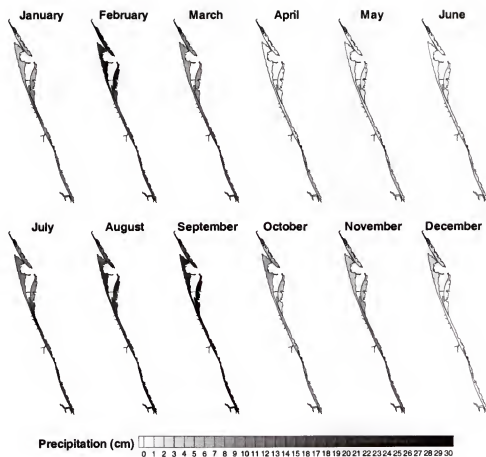
Figure I.1: Monthly precipitation into the IRL.
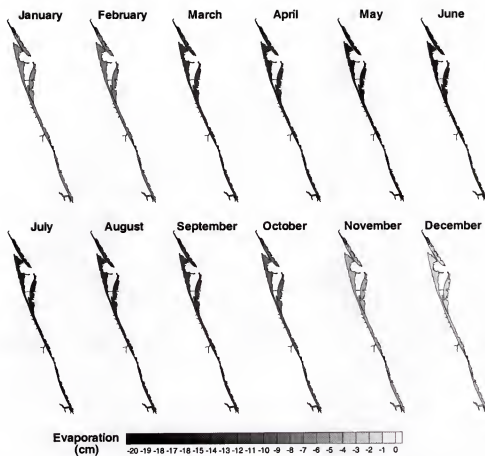
Figure I.2: Monthly evaporation rate based on historical Vero Beach data into the IRL.
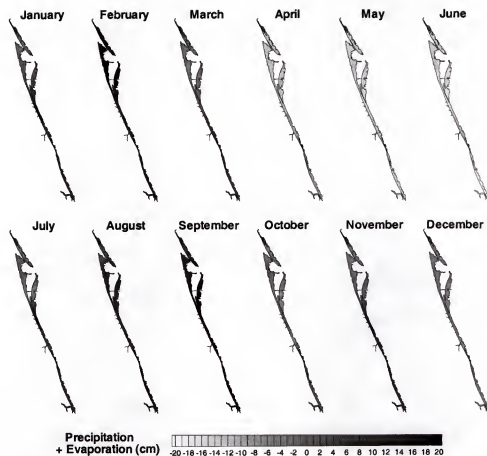
Figure I.3: Combined monthly precipitation and evaporation rates into the IRL. Evaporation rate is based on historical Vero Beach data.
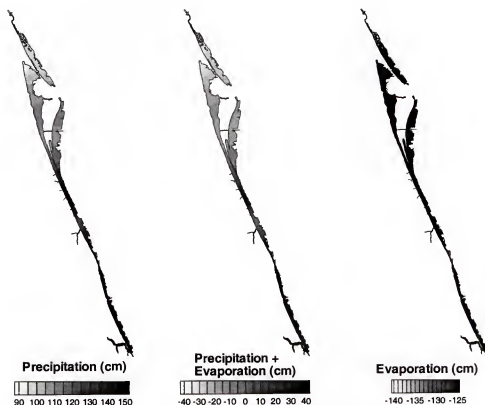
Figure I.4: Yearly totals of evaporation, precipitation and combined evaporation and precipitation into the IRL. Total evaporation is based on historical Vero Beach data.
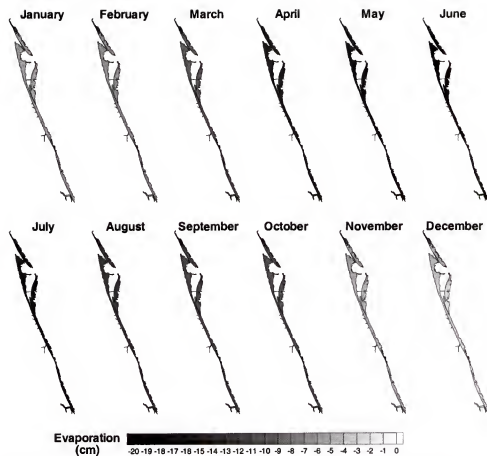
Figure I.5: Monthly evaporation rate based on the percentile formulation.
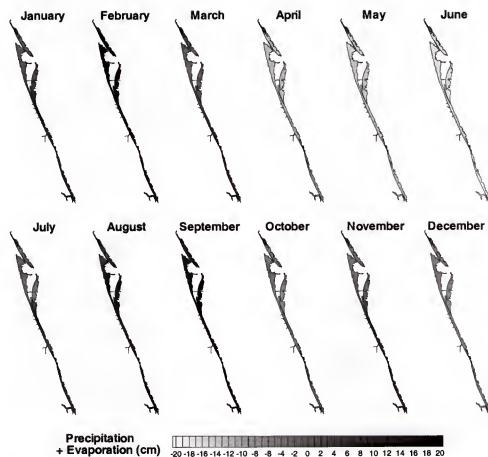
Figure I.6: Combined monthly precipitation and evaporation rates into the IRL. Evaporation rate is based on the percentile formulation.
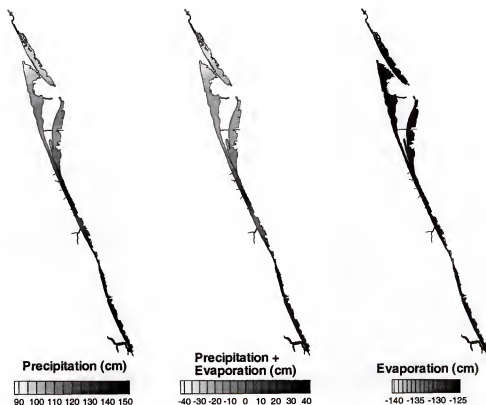
Figure I.7: Yearly totals of evaporation, precipitation and combined evaporation and precipitation into the IRL. Total evaporation is based on the percentile formulation.
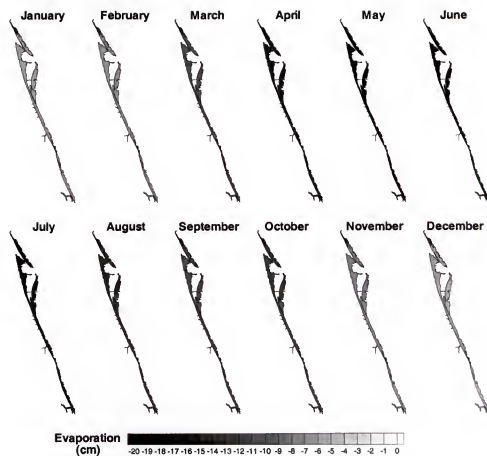
Figure I.8: Monthly evaporation rate based on the three station, spatially interpolated percentile formulation.
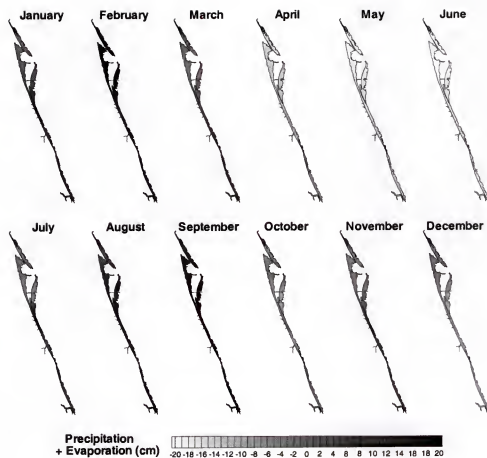
Figure I.9: Combined monthly precipitation and evaporation rates into the IRL. Evaporation rate is based on the the three station, spatially interpolated percentile formulation.
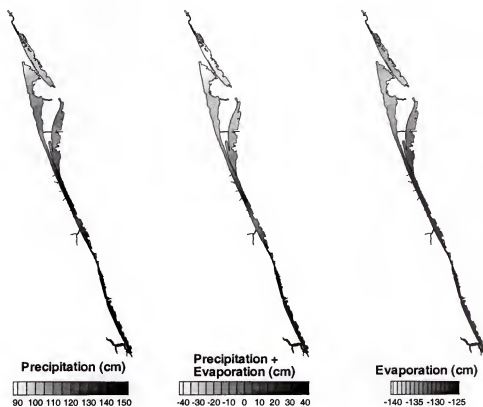
Figure I.10: Yearly totals of evaporation, precipitation and combined evaporation and precipitation into the IRL. Total evaporation is based on the three station, spatially interpolated percentile formulation.
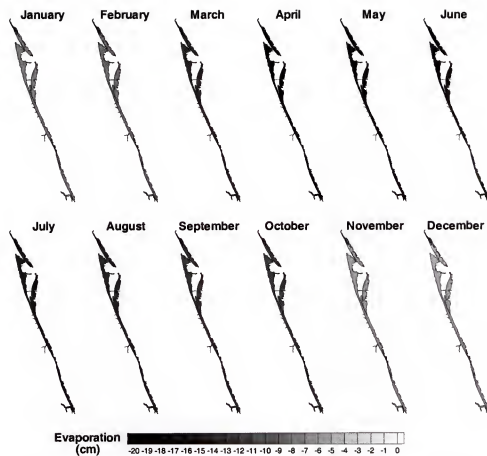
Figure I.11: Monthly evaporation rate based on Belle Glade data.
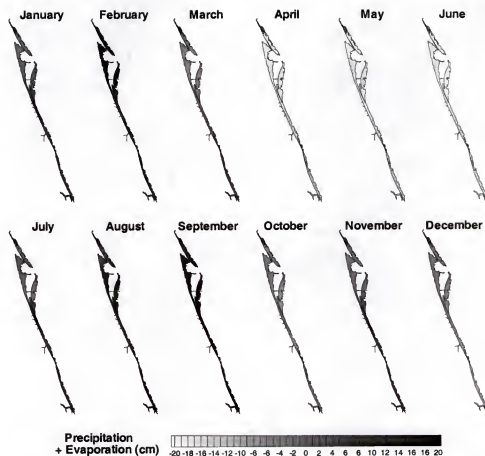
Figure I.12: Combined monthly precipitation and evaporation rates into the IRL. Evaporation rate is based on measured Belle Glade data.
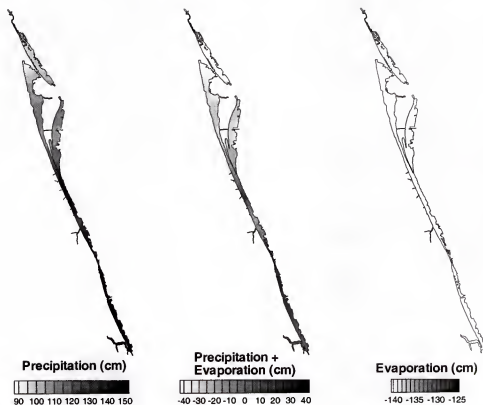
Figure I.13: Yearly totals of evaporation, precipitation and combined evaporation and precipitation into the IRL. Total evaporation is based on measured Belle Glade data.

APPENDIX J
WATER LEVEL COMPARISONS IN THE CH3D IRL FINE GRID

The following figures compare the simulated water level with the measured water level within the Indian River Lagoon in the fine grid. Figures J.1 through J.14 are year-long comparisons while Figures J.15 through J.28 compare only four days: February 14 (45), May 15 (135), August 13 (225) and November 11 (315). The stations used to compare are the Ponce Inlet Station (FDEP #872-1147), the Mosquito Lagoon Station (FDEP #872-1164), the Haulover Canal Station (USGS #02248380), the Titusville Station (FDEP #872-1456), the Merritt East Station (FDEP #872-1647), the Merritt West Station (FDEP #872-1648), the Banana River Station (FDEP #872-1789), the Melbourne Station (FDEP #872-1843), the Sebastian Inlet Station (FDEP #872-2004), the Wabasso Station (USGS #02251800), the Vero Bridge Station (FDEP #872-2125), the Fort Pierce Causeway Station (FDEP #872-2208), the Fort Pierce Inlet Station (FDEP #872-2213) and the St. Lucie Inlet Station (FDEP #872-2375)
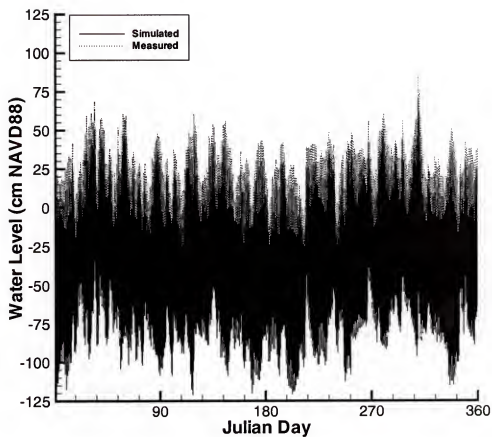
Figure J.1: A comparison between simulated and measured water level at the Ponce Inlet Station (FDEP #872-1147) during the entire 1998 simulation.

Figure J.2: A comparison between simulated and measured water level at the Mosquito Lagoon Station (FDEP #872-1164) during the entire 1998 simulation.

Figure J.3: A comparison between simulated and measured water level at the Haulover Canal Station (USGS #02248380) during the entire 1998 simulation.

Figure J.4: A comparison between simulated and measured water level at the Titusville Station (FDEP #872-1456) during the entire 1998 simulation.

Figure J.5: A comparison between simulated and measured water level at the Merritt East Station (FDEP #872-1647) during the entire 1998 simulation.

Figure J.6: A comparison between simulated and measured water level at the Merritt West Station (FDEP #872-1648) during the entire 1998 simulation.

Figure J.7: A comparison between simulated and measured water level at the Banana River Station (FDEP #872-1789) during the entire 1998 simulation.

Figure J.8: A comparison between simulated and measured water level at the Melbourne Station (FDEP #872-1843) during the entire 1998 simulation.

Figure J.9: A comparison between simulated and measured water level at the Sebastian Inlet Station (FDEP #872-2004) during the entire 1998 simulation.

Figure J.10: A comparison between simulated and measured water level at the Wabasso Station (USGS #02251800) during the entire 1998 simulation.

Figure J.11: A comparison between simulated and measured water level at the Vero Bridge Station (FDEP #872-2125) during the entire 1998 simulation.
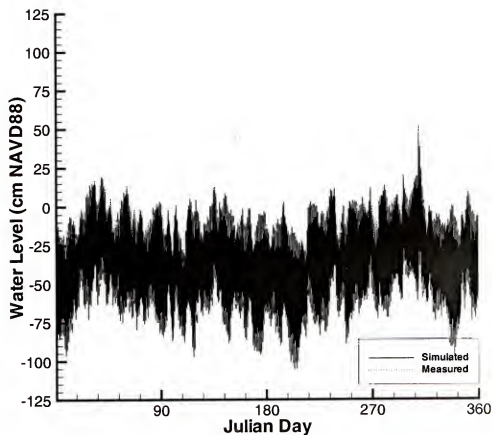
Figure J.12: A comparison between simulated and measured water level at the Fort Pierce Causeway Station (FDEP #872-2208) during the entire 1998 simulation.

Figure J.13: A comparison between simulated and measured water level at the Fort Pierce Inlet Station (FDEP #872-2213) during the entire 1998 simulation.

Figure J.14: A comparison between simulated and measured water level at the St. Lucie Inlet Station (FDEP #872-2375) during the entire 1998 simulation.
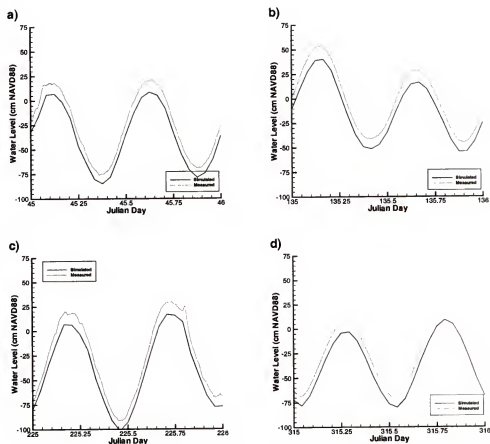
Figure J.15: A comparison between simulated and measured water level at the Ponce Inlet Station (FDEP #872-1147) during four days of the 1998 simulation: a) February 14 (45), b) May 15 (135), c) August 13 (225) and d) November 11 (315).
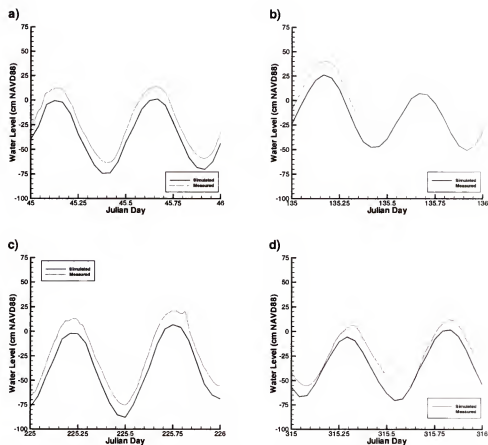
Figure J.16: A comparison between simulated and measured water level at the Mosquito Lagoon Station (FDEP #872-1164) during four days of the 1998 simulation: a) February 14 (45), b) May 15 (135), c) August 13 (225) and d) November 11 (315).
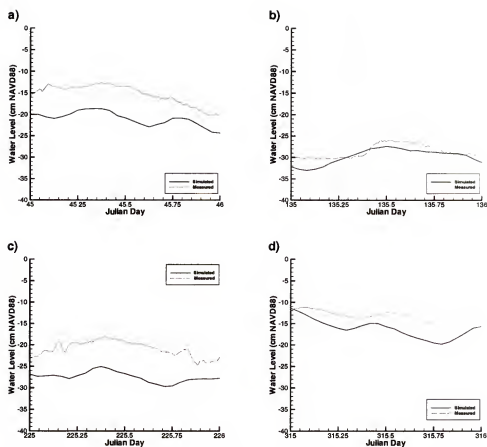
Figure J.17: A comparison between simulated and measured water level at the Haulover Canal Station (USGS #02248380) during four days of the 1998 simulation: a) February 14 (45), b) May 15 (135), c) August 13 (225) and d) November 11 (315).
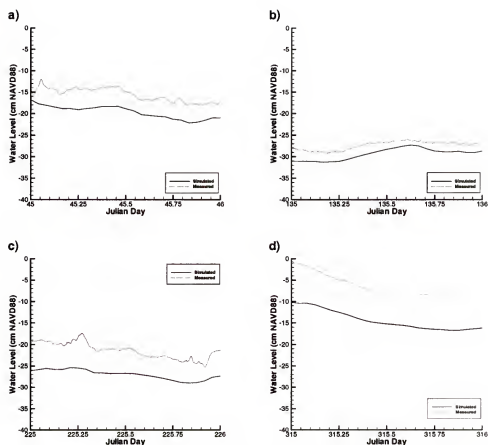
Figure J.18: A comparison between simulated and measured water level at the Titusville Station (FDEP #872-1456) during four days of the 1998 simulation: a) February 14 (45), b) May 15 (135), c) August 13 (225) and d) November 11 (315).
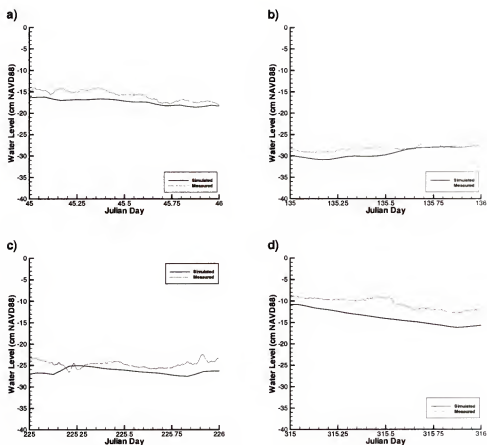
Figure J.19: A comparison between simulated and measured water level at the Merritt East Station (FDEP #872-1647) during four days of the 1998 simulation: a) February 14 (45), b) May 15 (135), c) August 13 (225) and d) November 11 (315).
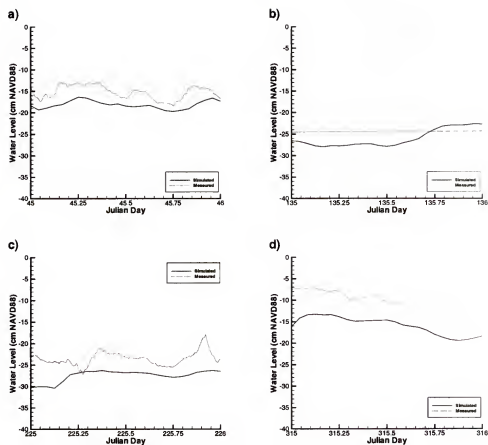
Figure J.20: A comparison between simulated and measured water level at the Merritt West Station (FDEP #872-1648) during four days of the 1998 simulation: a) February 14 (45), b) May 15 (135), c) August 13 (225) and d) November 11 (315).
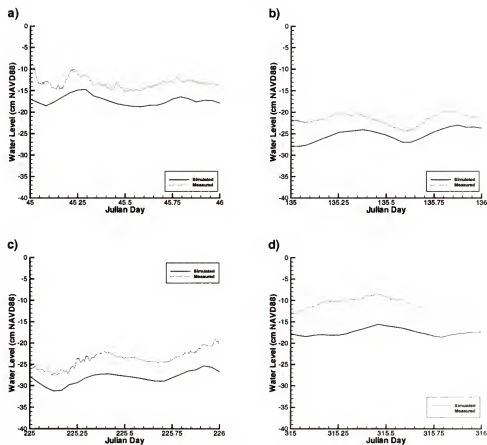
Figure J.21: A comparison between simulated and measured water level at the Banana River Station (FDEP #872-1789) during four days of the 1998 simulation: a) February 14 (45), b) May 15 (135), c) August 13 (225) and d) November 11 (315).
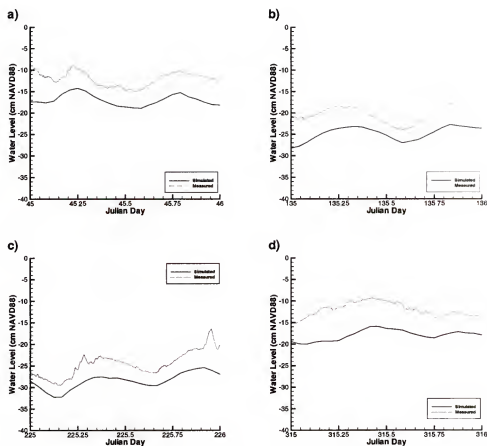
Figure J.22: A comparison between simulated and measured water level at the Melbourne Station (FDEP #872-1843) during four days of the 1998 simulation: a) February 14 (45), b) May 15 (135), c) August 13 (225) and d) November 11 (315).
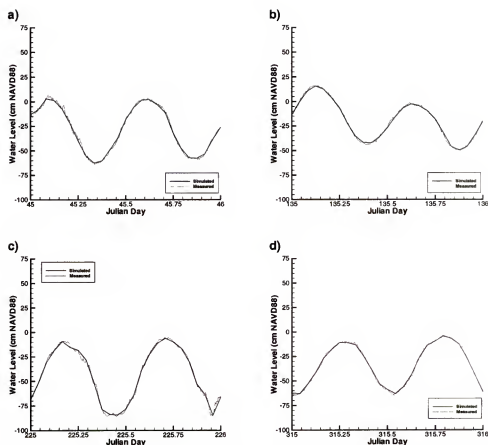
Figure J.23: A comparison between simulated and measured water level at the Sebastian Inlet Station (FDEP #872-2004) during four days of the 1998 simulation: a) February 14 (45), b) May 15 (135), c) August 13 (225) and d) November 11 (315).
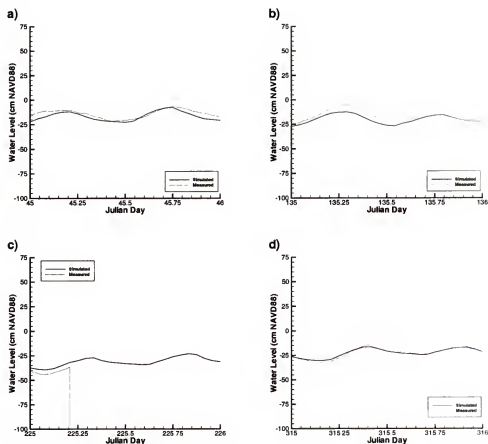
Figure J.24: A comparison between simulated and measured water level at the Wabasso Station (USGS #02251800) during four days of the 1998 simulation: a) February 14 (45), b) May 15 (135), c) August 13 (225) and d) November 11 (315).
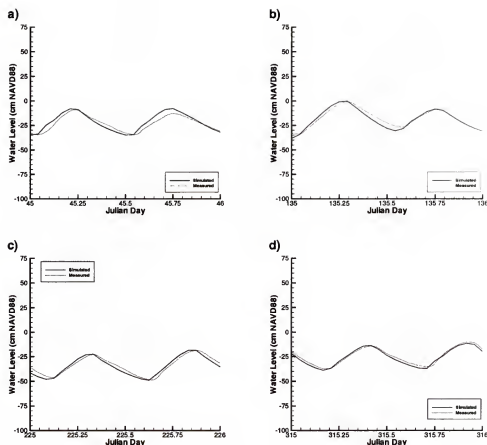
Figure J.25: A comparison between simulated and measured water level at the Vero Bridge Station (FDEP #872-2125) during four days of the 1998 simulation: a) February 14 (45), b) May 15 (135), c) August 13 (225) and d) November 11 (315).
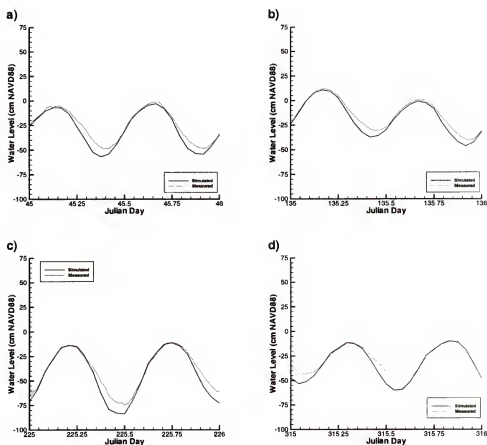
Figure J.26: A comparison between simulated and measured water level at the Fort Pierce Causeway Station (FDEP #872-2208) during four days of the 1998 simulation: a) February 14 (45), b) May 15 (135), c) August 13 (225) and d) November 11 (315).
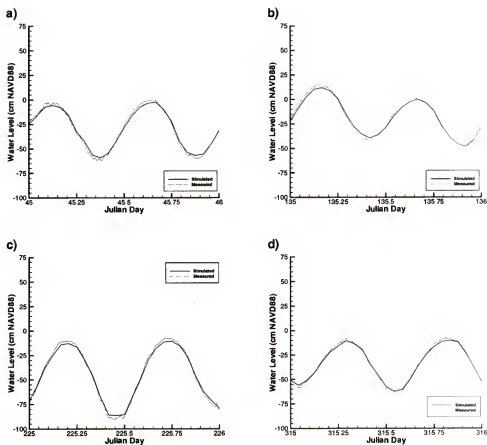
Figure J.27: A comparison between simulated and measured water level at the Fort Pierce Inlet Station (FDEP #872-2213) during four days of the 1998 simulation: a) February 14 (45), b) May 15 (135), c) August 13 (225) and d) November 11 (315).
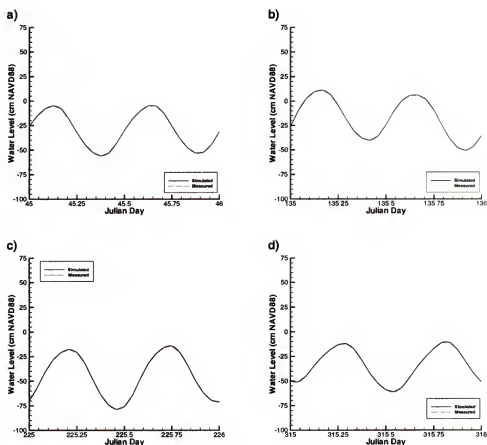
Figure J.28: A comparison between simulated and measured water level at the St. Lucie Inlet Station (FDEP #872-2375) during four days of the 1998 simulation: a) February 14 (45), b) May 15 (135), c) August 13 (225) and d) November 11 (315).

APPENDIX K
SPECTRAL DENSITY COMPARISONS IN THE CH3D IRL FINE GRID

Figures K.1 through K.14 compare simulated versus measured water level spectral density at the available water level stations for the entire 1998 simulation in the fine grid. The spectral density is calculated using MATLAB[1]. A 256 point FFT is used with a 256 point Hanning window. Any missing measured data was replaced with yearly mean values.

---

[1]http://www.mathworks.com
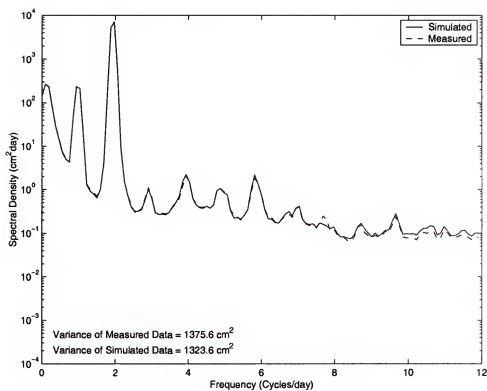
Figure K.1: A comparison between simulated and measured spectral density of water level at the Ponce Inlet Station (FDEP #872-1147) during the entire 1998 simulation.
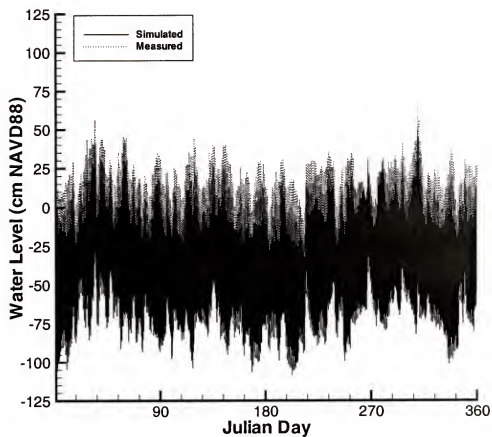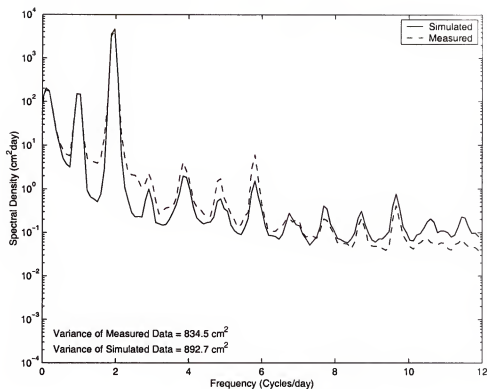
Figure K.2: A comparison between simulated and measured spectral density of water level at the Mosquito Lagoon Station (FDEP #872-1164) during the entire 1998 simulation.
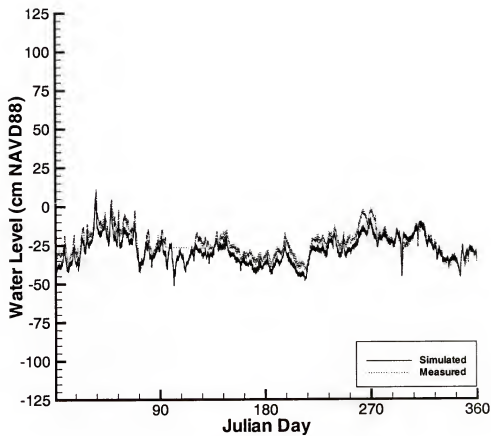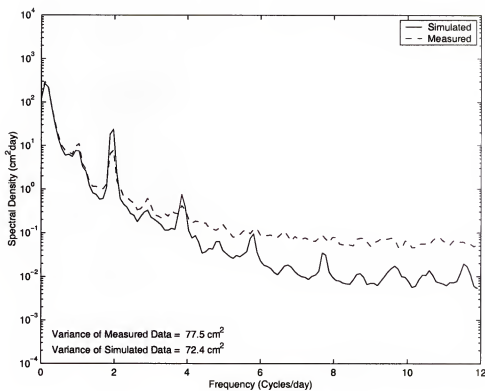
Figure K.3: A comparison between simulated and measured spectral density of water level at the Haulover Canal Station (USGS #02248380) during the entire 1998 simulation.
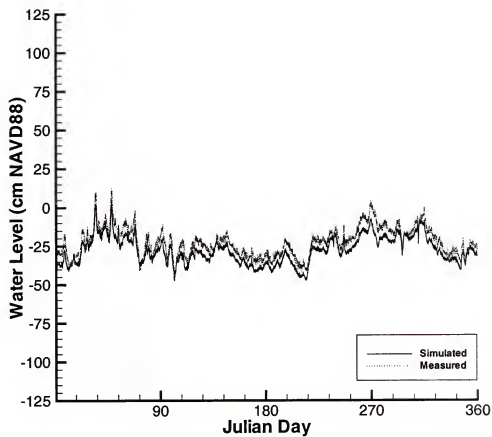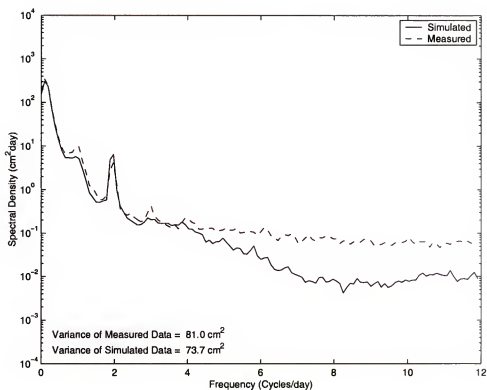
Figure K.4: A comparison between simulated and measured spectral density of water level at the Titusville Station (FDEP #872-1456) during the entire 1998 simulation.
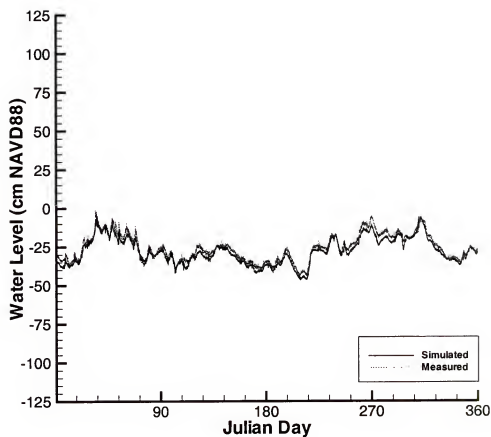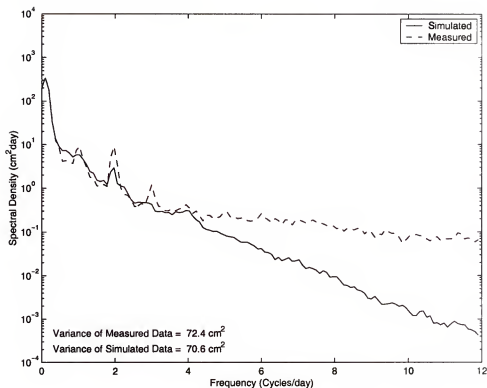
Figure K.5: A comparison between simulated and measured spectral density of water level at the Merritt East Station (FDEP #872-1647) during the entire 1998 simulation.
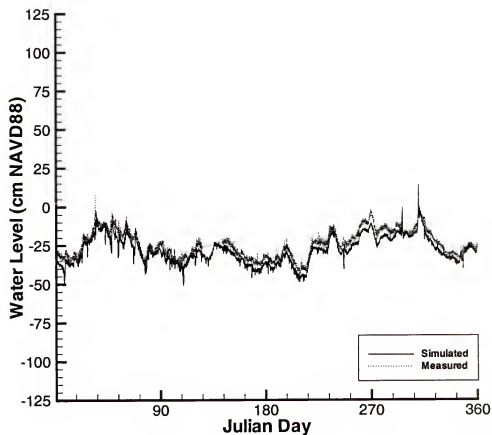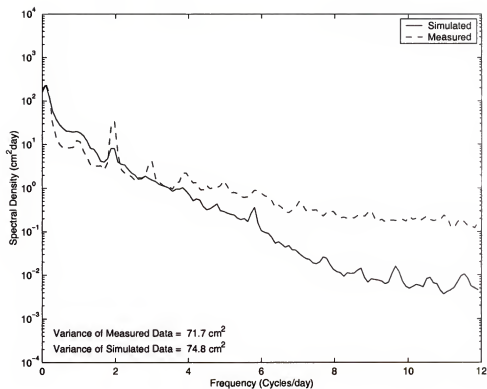
Figure K.6: A comparison between simulated and measured spectral density of water level at the Merritt West Station (FDEP #872-1648) during the entire 1998 simulation.
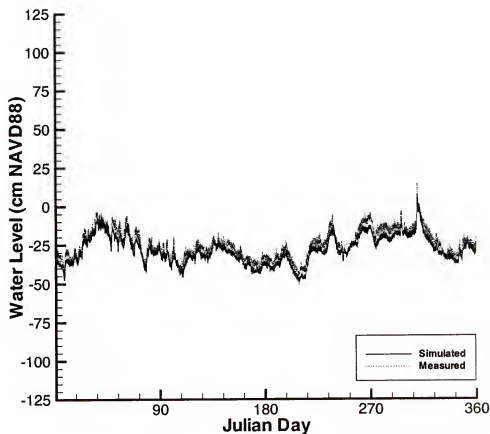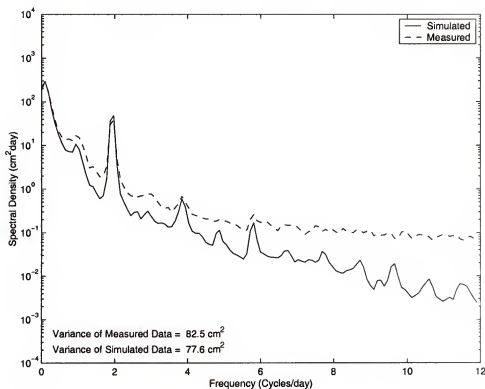
Figure K.7: A comparison between simulated and measured spectral density of water level at the Banana River Station (FDEP #872-1789) during the entire 1998 simulation.
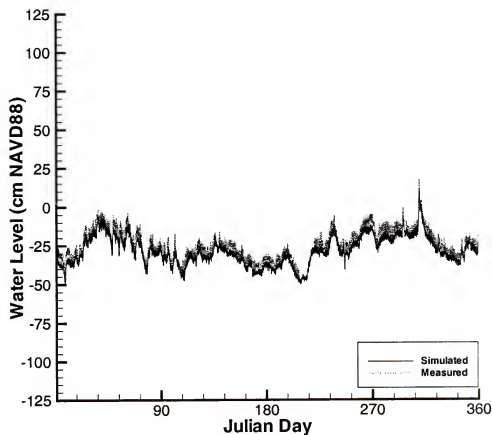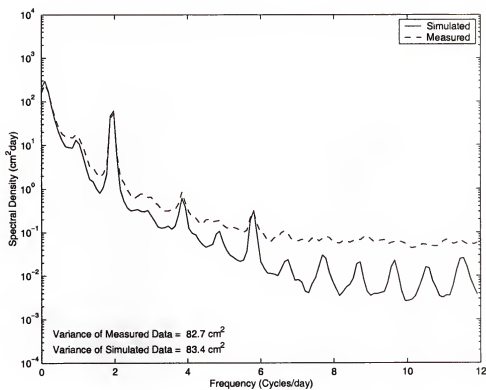
Figure K.8: A comparison between simulated and measured spectral density of water level at the Melbourne Station (FDEP #872-1843) during the entire 1998 simulation.
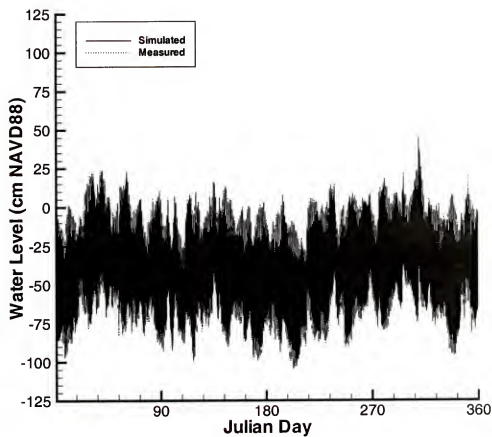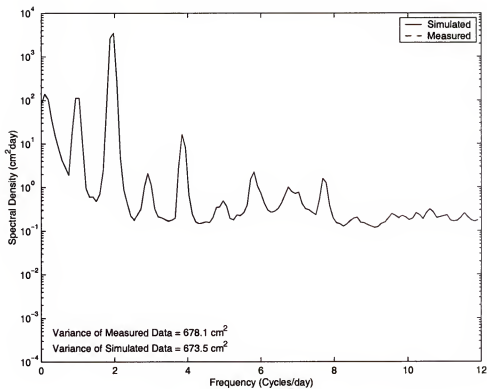
Figure K.9: A comparison between simulated and measured spectral density of water level at the Sebastian Inlet Station (FDEP #872-2004) during the entire 1998 simulation.
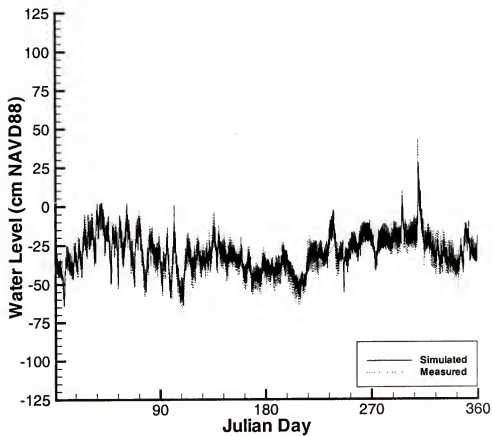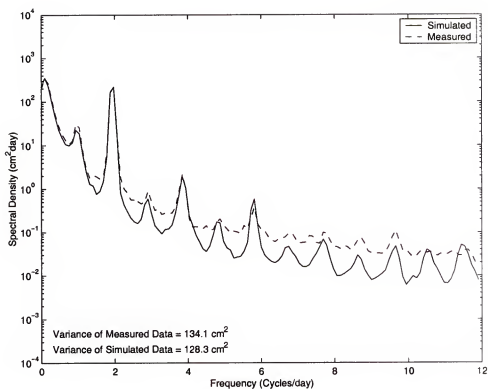
Figure K.10: A comparison between simulated and measured spectral density of water level at the Wabasso Station (USGS #02251800) during the entire 1998 simulation.
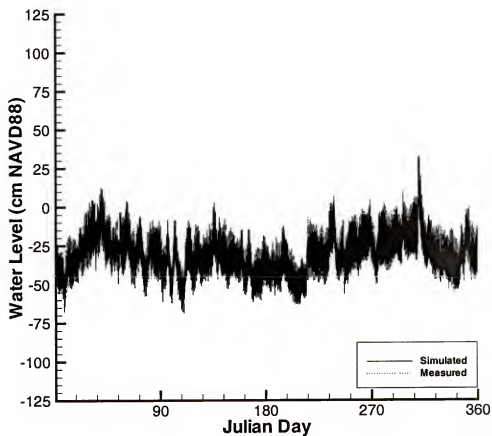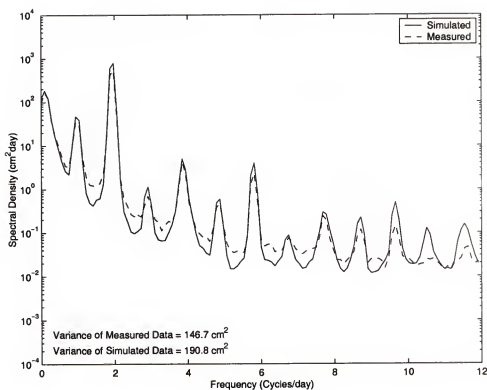
Figure K.11: A comparison between simulated and measured spectral density of water level at the Vero Bridge Station (FDEP #872-2125) during the entire 1998 simulation.
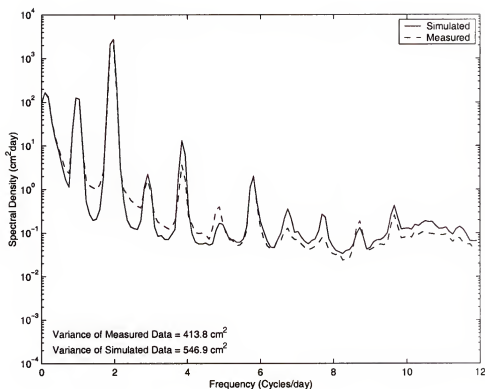
Figure.K.12: A comparison between simulated and measured spectral density of water level at the Fort Pierce Causeway Station (FDEP #872-2208) during the entire 1998 simulation.
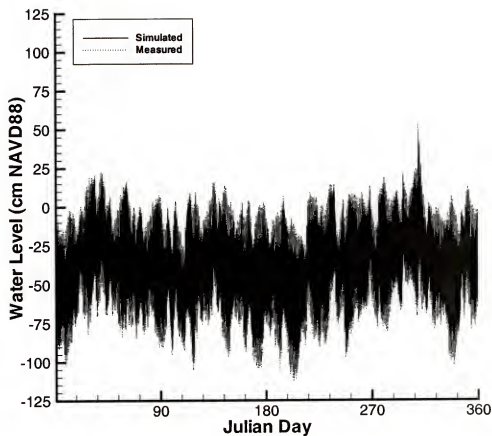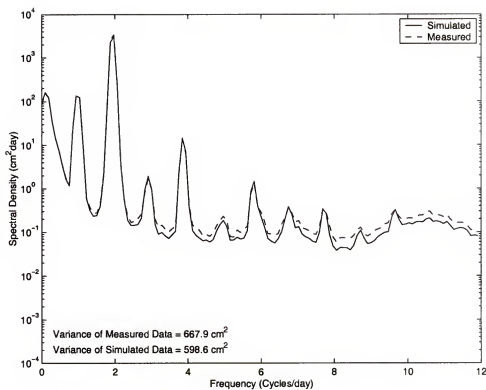
Figure K.13: A comparison between simulated and measured spectral density of water level at the Fort Pierce Inlet Station (FDEP #872-2213) during the entire 1998 simulation.
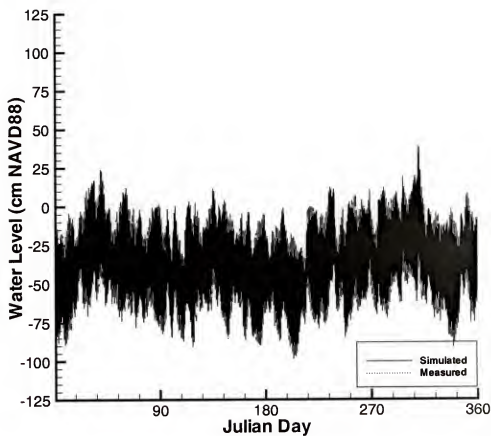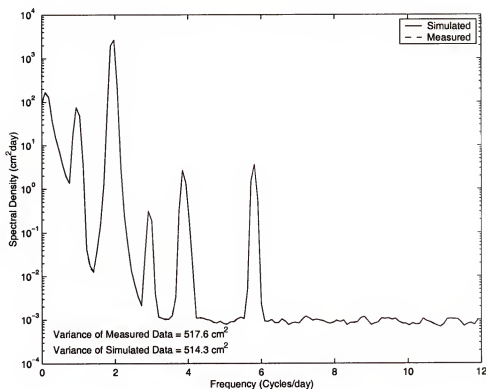
Figure K.14: A comparison between simulated and measured spectral density of water level at the St. Lucie Inlet Station (FDEP #872-2375) during the entire 1998 simulation.

APPENDIX L
FLOW RATE COMPARISONS IN THE CH3D IRL FINE GRID

The following figures compare the simulated flow rate with the USGS ADCP measured flow rate at the inlets located within the Indian River Lagoon domain in the fine grid. Figures L.1 through L.3 compare flow rates on four days during 1998 at Ponce de Leon Inlet, Sebastian Inlet and Ft. Pierce Inlet, respectively. Figure L.4 compares the simulated and measured flow rate at St. Lucie Inlet during two days of 1998.
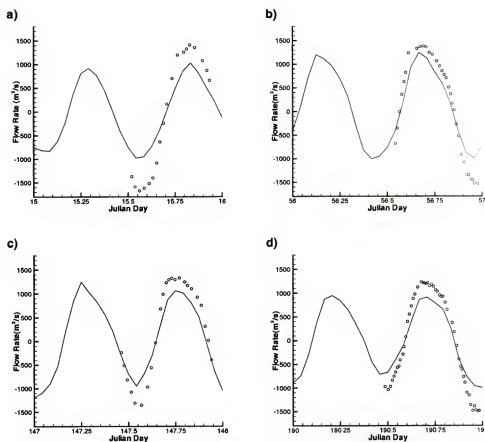
Figure L.1: A comparison between simulated and measured flow rate at Ponce de Leon Inlet. The simulated flow rate is shown with a solid line while the measured flow rate is shown with circles and a positive flow rate indicates flow from the lagoon into the Atlantic Ocean. The measured flow rates were measured by USGS ADCP four times during 1998: a) January 15 (15), b) February 25 (56), c) May 27 (147), and d) July 9 (190).
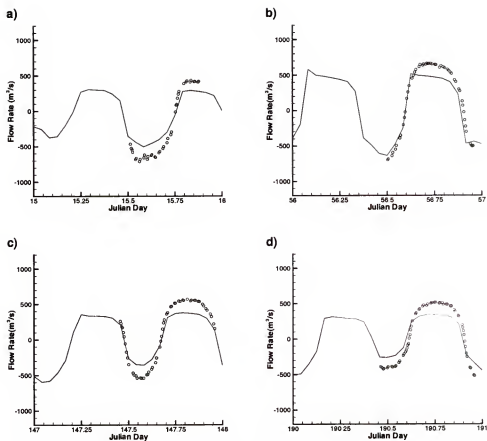
Figure L.2: A comparison between simulated and measured flow rate at Sebastian Inlet. The simulated flow rate is shown with a solid line while the measured flow rate is shown with circles and a positive flow rate indicates flow from the lagoon into the Atlantic Ocean. The measured flow rates were measured by USGS ADCP four times during 1998: a) January 15 (15), b) February 25 (56), c) May 27 (147), and d) July 9 (190).
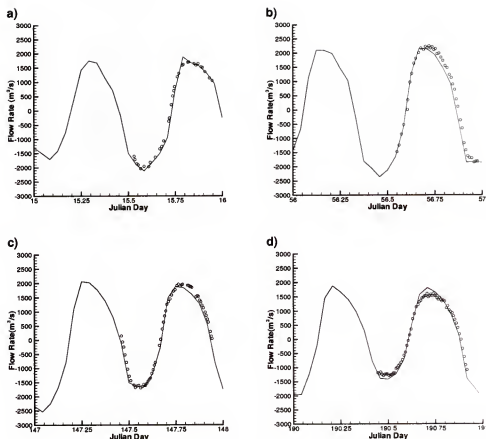
Figure L.3: A comparison between simulated and measured flow rate at Ft. Pierce Inlet. The simulated flow rate is shown with a solid line while the measured flow rate is shown with circles and a positive flow indicates flow from the lagoon into the Atlantic Ocean. The measured flow rates were measured by USGS ADCP four times during 1998: a) January 15 (15), b) February 25 (56), c) May 27 (147), and d) July 9 (190).

a)

b)
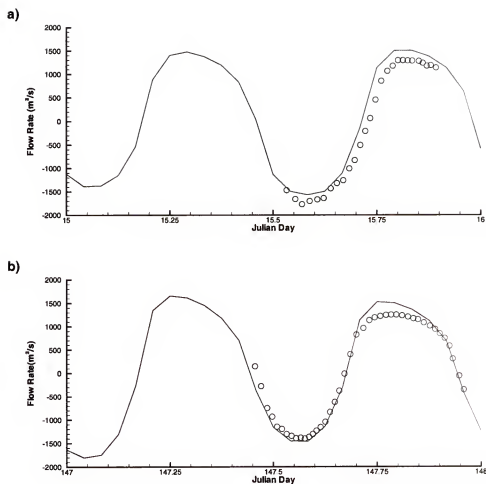
Figure L.4: A comparison between simulated and measured flow rate at St. Lucie. The simulated flow rate is shown with a solid line while the measured flow rate is shown with circles and a positive flow indicates flow from the lagoon into the Atlantic Ocean. The measured flow rates were measured by USGS ADCP two times during 1998: a) January 15 (15) and b) May 27 (147).

APPENDIX M
SALINITY COMPARISONS IN THE CH3D IRL FINE GRID

The following figures compare the simulated salinity with the measured salinity at both upper and lower levels within the Indian River Lagoon in the fine grid. The stations used to compare are the Mosquito Lagoon Station (FDEP #872-1164), the Southern Mosquito Lagoon Station (WQMN IRLML02), the Haulover Canal Station (USGS #02248380), the Titusville Station (FDEP #872-1456), the Merritt East Station (FDEP #872-1647), the Merritt West Station (FDEP #872-1648), the Banana River Station (FDEP #872-1789), the Melbourne Causeway Station (FDEP #872-1843), the Sebastian River Station (USGS #275017080295600), the Vero Bridge Station (FDEP #872-2125) and the Fort Pierce Causeway Station (FDEP #872-2208). The simulated salinity in the lower most vertical cell (12.5% of the water depth for four vertical layers) is used to compare with the lower measured salinity sensor and the upper measured salinity sensor is compared with an average of the salinity in the second and third vertical cells (50% of the water depth for four vertical layers).
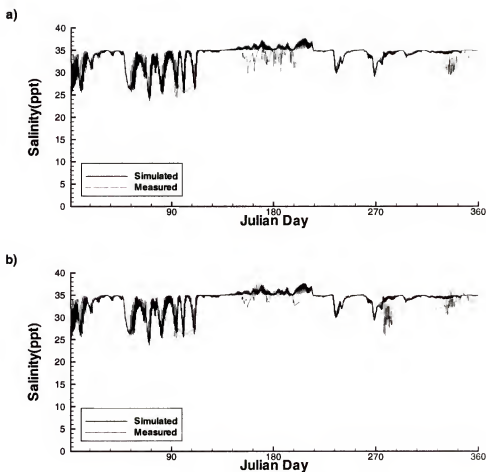
Figure M.1: A comparison between simulated and measured salinity at the Mosquito Lagoon Station (FDEP #872-1164). The upper level comparison is shown in a) and the lower level comparison in b).
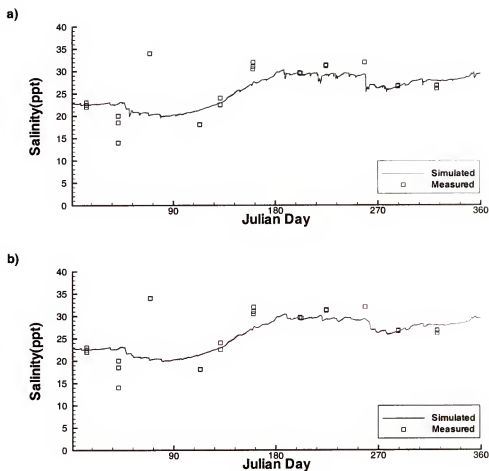
Figure M.2: A comparison between simulated and measured salinity at the Southern Mosquito Lagoon Station (WQMN IRLML02). The upper level comparison is shown in a) and the lower level comparison in b).
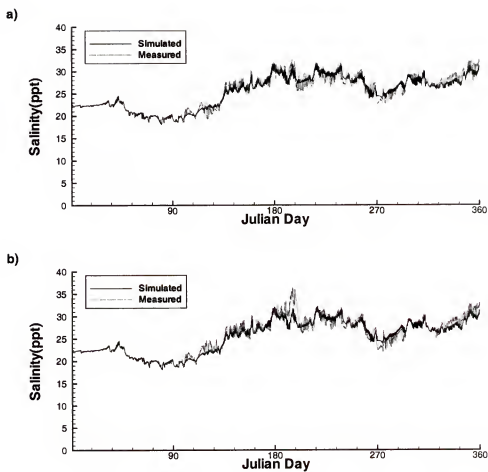
Figure M.3: A comparison between simulated and measured salinity at the Haulover Canal Station (USGS #02248380). The upper level comparison is shown in a) and the lower level comparison in b).
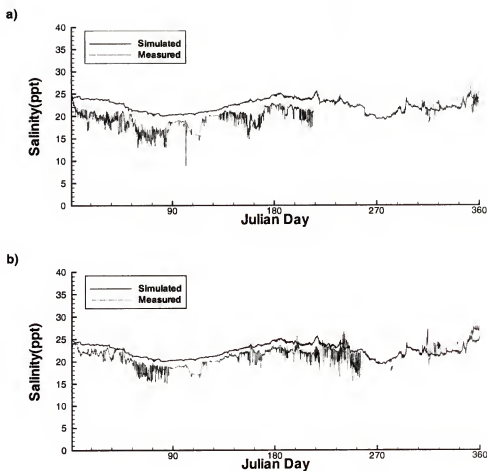
Figure M.4: A comparison between simulated and measured surface at the Titusville Station (FDEP #872-1456). The upper level comparison is shown in a) and the lower level comparison in b).
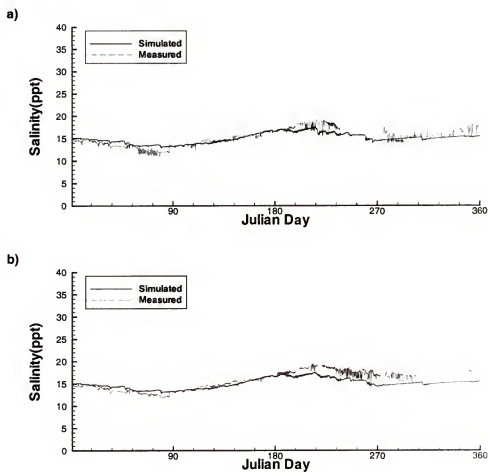
Figure M.5: A comparison between simulated and measured salinity at the Merritt East Station (FDEP #872-1647). The upper level comparison is shown in a) and the lower level comparison in b).
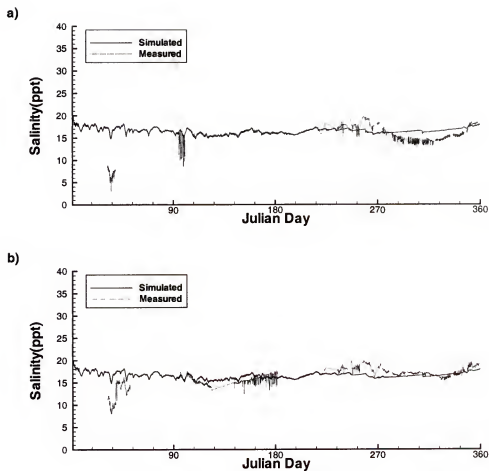
Figure M.6: A comparison between simulated and measured salinity at the Merritt West Station (FDEP #872-1648). The upper level comparison is shown in a) and the lower level comparison in b).
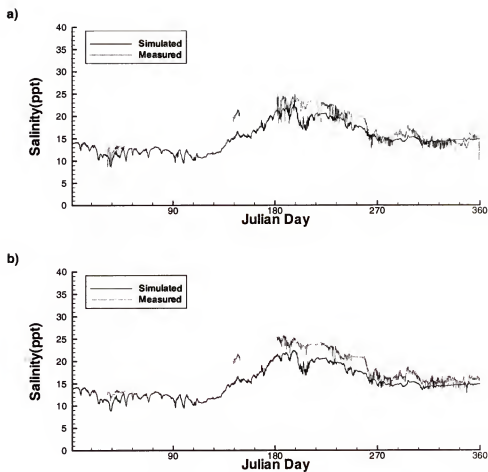
Figure M.7: A comparison between simulated and measured salinity at the Banana River Station (FDEP #872-1789). The upper level comparison is shown in a) and the lower level comparison is shown in b).
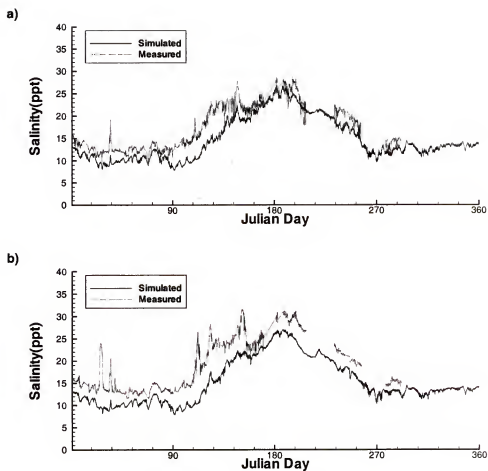
Figure M.8: A comparison between simulated and measured salinity at the Melbourne Causeway Station (FDEP #872-1843). The upper level comparison is shown in a) and the lower level comparison in b).
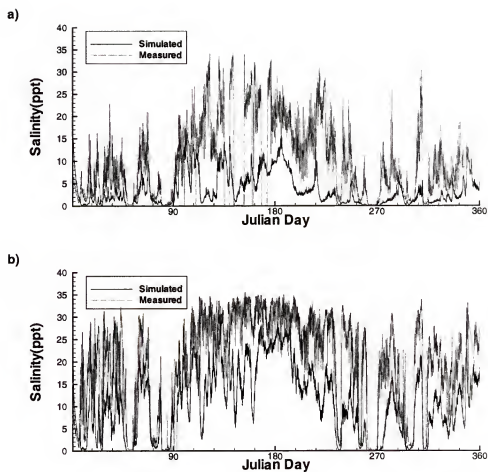
Figure M.9: A comparison between simulated and measured salinity at the Sebastian River Station (USGS #275017080295600). The upper level comparison is shown in a) and the lower level comparison in b).
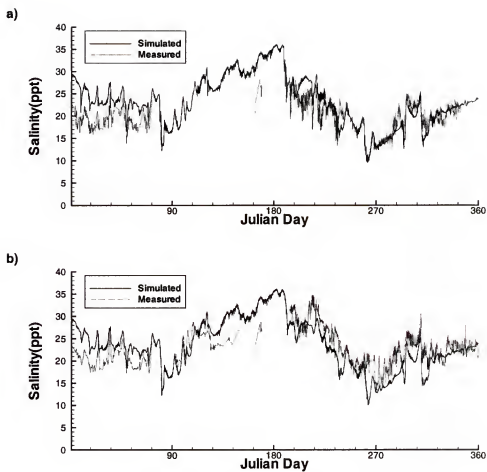
Figure M.10: A comparison between simulated and measured salinity at the Vero Bridge Station (FDEP #872-2125). The upper level comparison is shown in a) and the lower level comparison in b).
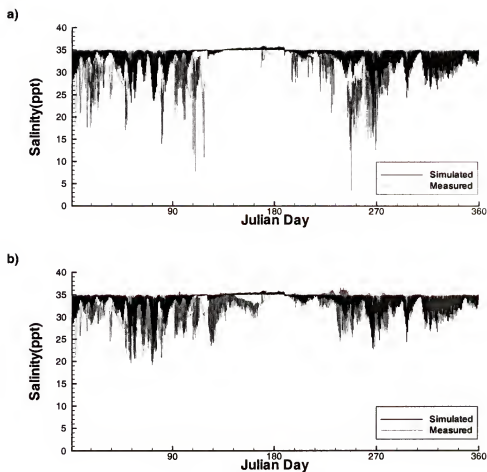
Figure M.11: A comparison between simulated and measured salinity at the Fort Pierce Causeway Station (FDEP #872-2208). The upper level comparison is shown in a) and the lower level comparison in b).

The following tables give the CPU times per iteration (and their associated speedups) for the parallel CH3D model. Tables N.1 through N.3 are for the hydrodynamic and salinity model. Tables N.4 through N.6 are for the hydrodynamic, salinity and flushing models. Tables N.7 through N.9 are for the hydrodynamic, salinity, sediment and water quality models applied to the "fine grid" while Tables N.10 through N.12 are for the same models applied to the "coarse grid".

Table N.1: CPU times for the parallel, shared memory, CH3D procedures which are used in a hydrodynamic and salinity simulation on the "ocean" computer. Times shown are per time step iteration of the model using the boundary-fitted "fine grid" (477x43) and and are given in seconds, n is the number of processors used and speedup is shown in parenthesis.

Procedure	Serial	$n = 1$	$n = 2$	$n = 3$	$n = 4$
Turbulence	0.069(1.00)	0.078(0.88)	0.037(1.85)	0.024(2.84)	0.018(3.72)
N.L./Diffusion (J)	0.063(1.00)	0.064(1.00)	0.034(1.88)	0.023(2.79)	0.017(3.64)
Baroclinic (I)	0.063(1.00)	0.062(1.01)	0.034(1.83)	0.023(2.69)	0.018(3.41)
Baroclinic (J)	0.054(1.00)	0.060(0.91)	0.031(1.74)	0.022(2.50)	0.017(3.20)
N.L./Diffusion (I)	0.047(1.00)	0.047(1.00)	0.026(1.77)	0.019(2.47)	0.015(3.05)
Salinity	0.039(1.00)	0.042(0.92)	0.026(1.51)	0.020(1.93)	0.017(2.22)
Layer Vel.(v)	0.038(1.00)	0.040(0.97)	0.022(1.76)	0.015(2.60)	0.011(3.36)
Layer Vel.(u)	0.025(1.00)	0.025(0.98)	0.013(1.86)	0.009(2.76)	0.007(3.64)
Interpolation	0.022(1.00)	0.023(1.00)	0.013(1.75)	0.009(2.40)	0.008(2.92)
Integrated Vel. (U)	0.016(1.00)	0.016(1.01)	0.012(1.35)	0.010(1.57)	0.009(1.73)
Dimensionalize	0.014(1.00)	0.014(0.98)	0.008(1.75)	0.005(2.64)	0.004(3.23)
Layer Vel.(w)	0.013(1.00)	0.013(0.99)	0.008(1.72)	0.005(2.51)	0.004(3.20)
Integrate. Vel. (V)	0.013(1.00)	0.013(0.96)	0.008(1.61)	0.006(2.26)	0.005(2.82)
All Parallel Routines	0.475(1.00)	0.496(0.96)	0.271(1.75)	0.190(2.50)	0.152(3.13)
Total Runtime	0.502(1.00)	0.522(0.96)	0.298(1.68)	0.217(2.31)	0.179(2.80)

Table N.2: CPU times for the parallel, shared memory, CH3D procedures which are used in a hydrodynamic and salinity simulation on the "seiche" computer. Times shown are per time step iteration of the model using the boundary-fitted "fine grid" (477x43) and and are given in seconds, n is the number of processors used and speedup is shown in parenthesis.

Procedure	Serial	$n = 1$	$n = 2$
Turbulence	0.065(1.00)	0.073(0.88)	0.034(1.89)
Baroclinic (I)	0.059(1.00)	0.059(0.99)	0.031(1.88)
N.L./Diffusion (J)	0.058(1.00)	0.058(1.00)	0.030(1.93)
Baroclinic (J)	0.052(1.00)	0.056(0.92)	0.028(1.86)
N.L./Diffusion (I)	0.044(1.00)	0.043(1.02)	0.024(1.86)
Salinity	0.039(1.00)	0.042(0.93)	0.024(1.62)
Layer Vel.(v)	0.036(1.00)	0.038(0.96)	0.020(1.85)
Layer Vel.(u)	0.024(1.00)	0.024(0.98)	0.012(1.97)
Interpolation	0.021(1.00)	0.021(1.00)	0.012(1.77)
Integrated Vel. (U)	0.016(1.00)	0.016(1.00)	0.011(1.44)
Layer Vel.(w)	0.013(1.00)	0.013(1.00)	0.007(1.92)
Integrate. Vel. (V)	0.012(1.00)	0.013(0.97)	0.007(1.73)
Dimensionalize	0.012(1.00)	0.012(0.99)	0.007(1.78)
All Parallel Routines	0.452(1.00)	0.469(0.96)	0.247(1.83)
Total Runtime	0.480(1.00)	0.496(0.97)	0.274(1.75)

Table N.3: CPU times for the parallel, shared memory, CH3D procedures which are used in a hydrodynamic and salinity simulation on the "nereus" computer. Times shown are per time step iteration of the model using the boundary-fitted "fine grid" (477x43) and and are given in seconds, n is the number of processors used and speedup is shown in parenthesis.

Procedure	Serial	$n = 1$	$n = 2$
Turbulence	0.071(1.00)	0.071(1.01)	0.036(1.96)
Baroclinic (I)	0.052(1.00)	0.053(0.99)	0.028(1.90)
N.L./Diffusion (I)	0.045(1.00)	0.044(1.00)	0.024(1.85)
N.L./Diffusion (J)	0.044(1.00)	0.044(1.00)	0.023(1.97)
Baroclinic (J)	0.044(1.00)	0.043(1.01)	0.023(1.93)
Salinity	0.037(1.00)	0.035(1.04)	0.021(1.74)
Interpolation	0.029(1.00)	0.029(1.02)	0.015(1.89)
Layer Vel.(v)	0.025(1.00)	0.025(0.99)	0.013(1.93)
Layer Vel.(u)	0.021(1.00)	0.021(1.00)	0.010(1.96)
Layer Vel.(w)	0.015(1.00)	0.015(0.99)	0.008(1.95)
Integrated Vel. (U)	0.015(1.00)	0.015(0.99)	0.010(1.48)
Integrate. Vel. (V)	0.011(1.00)	0.011(0.99)	0.006(1.91)
Dimensionalize	0.009(1.00)	0.010(0.99)	0.005(1.77)
All Parallel Routines	0.418(1.00)	0.416(1.01)	0.222(1.88)
Total Runtime	0.438(1.00)	0.436(1.00)	0.242(1.81)

Table N.4: CPU times for the parallel, shared memory, CH3D procedures which are used in a hydrodynamic, salinity and flushing simulation on the "ocean" computer. Times shown are per time step iteration of the model using the boundary-fitted "fine grid" (477x43) and and are given in seconds, n is the number of processors used and speedup is shown in parenthesis.

Procedure	Serial	$n = 1$	$n = 2$	$n = 3$	$n = 4$
Species	0.258(1.00)	0.286(0.90)	0.149(1.73)	0.114(2.26)	0.097(2.66)
Turbulence	0.071(1.00)	0.076(0.93)	0.038(1.88)	0.025(2.86)	0.019(3.68)
Baroclinic (I)	0.064(1.00)	0.068(0.95)	0.035(1.82)	0.025(2.61)	0.020(3.20)
N.L./Diffusion (J)	0.060(1.00)	0.063(0.94)	0.032(1.86)	0.022(2.72)	0.017(3.52)
Baroclinic (J)	0.056(1.00)	0.071(0.78)	0.034(1.62)	0.024(2.32)	0.019(2.91)
N.L./Diffusion (I)	0.047(1.00)	0.051(0.91)	0.026(1.80)	0.019(2.48)	0.015(3.07)
Layer Vel.(v)	0.039(1.00)	0.041(0.96)	0.021(1.85)	0.015(2.68)	0.011(3.43)
Salinity	0.038(1.00)	0.042(0.91)	0.025(1.52)	0.020(1.95)	0.017(2.25)
Layer Vel.(u)	0.024(1.00)	0.026(0.94)	0.013(1.84)	0.009(2.72)	0.007(3.55)
Interpolation	0.023(1.00)	0.024(0.94)	0.016(1.45)	0.011(2.02)	0.009(2.45)
Integrated Vel. (U)	0.015(1.00)	0.016(0.97)	0.012(1.25)	0.011(1.45)	0.010(1.61)
Layer Vel.(w)	0.013(1.00)	0.013(1.00)	0.008(1.74)	0.005(2.50)	0.004(3.22)
Dimensionalize	0.013(1.00)	0.015(0.87)	0.008(1.51)	0.006(2.29)	0.004(2.83)
Integrate. Vel. (V)	0.013(1.00)	0.013(0.98)	0.008(1.59)	0.006(2.27)	0.005(2.79)
All Parallel Routines	0.733(1.00)	0.805(0.91)	0.426(1.72)	0.310(2.37)	0.255(2.88)
Total Runtime	0.768(1.00)	0.838(0.92)	0.461(1.67)	0.345(2.22)	0.291(2.64)

Table N.5: CPU times for the parallel, shared memory, CH3D procedures which are used in a hydrodynamic, salinity and flushing simulation on the "seiche" computer. Times shown are per time step iteration of the model using the boundary-fitted "fine grid" (477x43) and and are given in seconds, n is the number of processors used and speedup is shown in parenthesis.

Procedure	Serial	$n = 1$	$n = 2$
Species	0.245(1.00)	0.258(0.95)	0.137(1.79)
Turbulence	0.066(1.00)	0.074(0.90)	0.035(1.90)
Baroclinic (I)	0.060(1.00)	0.061(0.99)	0.031(1.91)
N.L./Diffusion (J)	0.054(1.00)	0.057(0.95)	0.029(1.89)
Baroclinic (J)	0.053(1.00)	0.056(0.94)	0.028(1.91)
N.L./Diffusion (I)	0.044(1.00)	0.044(1.01)	0.023(1.89)
Salinity	0.039(1.00)	0.041(0.95)	0.024(1.64)
Layer Vel.(v)	0.036(1.00)	0.038(0.96)	0.019(1.94)
Layer Vel.(u)	0.024(1.00)	0.024(0.98)	0.012(1.97)
Interpolation	0.021(1.00)	0.021(1.00)	0.012(1.75)
Integrated Vel. (U)	0.016(1.00)	0.016(0.99)	0.011(1.43)
Integrate. Vel. (V)	0.014(1.00)	0.014(0.98)	0.008(1.82)
Layer Vel.(w)	0.013(1.00)	0.013(0.99)	0.007(1.94)
Dimensionalize	0.011(1.00)	0.013(0.89)	0.007(1.62)
All Parallel Routines	0.697(1.00)	0.730(0.95)	0.382(1.82)
Total Runtime	0.734(1.00)	0.767(0.96)	0.417(1.76)

Table N.6: CPU times for the parallel, shared memory, CH3D procedures which are used in a hydrodynamic, salinity and flushing simulation on the "nereus" computer. Times shown are per time step iteration of the model using the boundary-fitted "fine grid" (477x43) and and are given in seconds, n is the number of processors used and speedup is shown in parenthesis.

Procedure	Serial	$n = 1$	$n = 2$
Species	0.266(1.00)	0.249(1.07)	0.143(1.86)
Turbulence	0.071(1.00)	0.071(1.00)	0.037(1.91)
Baroclinic (I)	0.052(1.00)	0.052(0.99)	0.028(1.87)
N.L./Diffusion (I)	0.045(1.00)	0.045(0.99)	0.024(1.84)
N.L./Diffusion (J)	0.044(1.00)	0.045(0.99)	0.023(1.94)
Baroclinic (J)	0.043(1.00)	0.043(1.00)	0.022(1.92)
Salinity	0.037(1.00)	0.035(1.04)	0.021(1.73)
Interpolation	0.029(1.00)	0.029(1.01)	0.016(1.88)
Layer Vel.(v)	0.025(1.00)	0.026(0.98)	0.013(1.91)
Layer Vel.(u)	0.020(1.00)	0.020(1.00)	0.010(1.96)
Layer Vel.(w)	0.015(1.00)	0.015(0.99)	0.008(1.95)
Integrated Vel. (U)	0.015(1.00)	0.015(1.01)	0.010(1.51)
Integrate. Vel. (V)	0.011(1.00)	0.011(0.99)	0.006(1.92)
Dimensionalize	0.009(1.00)	0.009(1.01)	0.005(1.76)
All Parallel Routines	0.682(1.00)	0.665(1.02)	0.366(1.86)
Total Runtime	0.706(1.00)	0.689(1.02)	0.390(1.81)

Table N.7: CPU times for the parallel, shared memory, CH3D procedures which are used in a hydrodynamic, salinity, sediment and nutrient simulation on the "ocean" computer. Times shown are per time step iteration of the model using the boundary-fitted "fine grid" (477x43) and are given in seconds, n is the number of processors used and speedup is shown in parenthesis. Times for procedures marked with a "*" are included with other procedures are not thus not included in the totals.

Procedure	Serial	$n = 1$	$n = 2$	$n = 3$	$n = 4$
Main WQ	1.309(1.00)	1.298(1.01)	0.699(1.87)	0.514(2.55)	0.413(3.17)
Nutrient Transport*	0.355(1.00)	0.375(0.95)	0.187(1.90)	0.138(2.57)	0.111(3.19)
Main Sediment	0.142(1.00)	0.152(0.94)	0.084(1.69)	0.066(2.17)	0.056(2.54)
Turbulence	0.070(1.00)	0.076(0.91)	0.038(1.85)	0.025(2.79)	0.019(3.62)
Baroclinic (I)	0.064(1.00)	0.067(0.94)	0.034(1.87)	0.024(2.65)	0.019(3.34)
Baroclinic (J)	0.064(1.00)	0.072(0.89)	0.033(1.90)	0.024(2.65)	0.019(3.42)
N.L./Diffusion (J)	0.062(1.00)	0.068(0.91)	0.034(1.82)	0.024(2.62)	0.018(3.45)
Fine Sediment*	0.051(1.00)	0.056(0.92)	0.027(1.93)	0.019(2.68)	0.015(3.43)
Coarse Sediment*	0.048(1.00)	0.051(0.95)	0.025(1.96)	0.018(2.74)	0.014(3.51)
N.L./Diffusion (I)	0.048(1.00)	0.052(0.93)	0.026(1.86)	0.019(2.53)	0.015(3.12)
Layer Vel.(v)	0.039(1.00)	0.040(0.97)	0.021(1.85)	0.015(2.67)	0.011(3.45)
Salinity	0.039(1.00)	0.042(0.92)	0.025(1.57)	0.020(1.98)	0.017(2.30)
Interpolation	0.030(1.00)	0.030(1.00)	0.018(1.61)	0.013(2.22)	0.011(2.73)
Layer Vel.(u)	0.025(1.00)	0.026(0.95)	0.013(1.91)	0.009(2.74)	0.007(3.63)
Dimensionalize	0.017(1.00)	0.016(1.04)	0.009(1.80)	0.007(2.54)	0.005(3.19)
Integrated Vel. (U)	0.016(1.00)	0.016(1.01)	0.012(1.35)	0.010(1.55)	0.010(1.69)
Layer Vel.(w)	0.013(1.00)	0.013(1.00)	0.008(1.77)	0.005(2.50)	0.004(3.27)
Integrate. Vel. (V)	0.013(1.00)	0.013(1.04)	0.008(1.65)	0.006(2.19)	0.005(2.75)
Bottom Shear Stress*	0.001(1.00)	0.001(1.03)	0.000(1.95)	0.000(2.85)	0.000(3.61)
Wave H/T*	0.001(1.00)	0.000(1.04)	0.000(2.04)	0.000(3.04)	0.000(3.95)
All Parallel Routines	1.950(1.00)	1.981(0.98)	1.062(1.84)	0.780(2.50)	0.629(3.10)
Total Runtime	1.958(1.00)	1.989(0.98)	1.078(1.82)	0.799(2.45)	0.650(3.01)

Table N.8: CPU times for the parallel, shared memory, CH3D procedures which are used in a hydrodynamic, salinity, sediment and nutrient simulation on the "seiche" computer. Times shown are per time step iteration of the model using the boundary-fitted "fine grid" (477x43) and are given in seconds, n is the number of processors used and speedup is shown in parenthesis. Times for procedures marked with a "*" are included with other procedures are not thus not included in the totals.

Procedure	Serial	$n = 1$	$n = 2$
Main WQ	1.222(1.00)	1.207(1.01)	0.641(1.91)
Nutrient Transport*	0.359(1.00)	0.378(0.95)	0.179(2.01)
Main Sediment	0.137(1.00)	0.147(0.93)	0.081(1.70)
Turbulence	0.066(1.00)	0.072(0.91)	0.035(1.86)
Baroclinic (J)	0.060(1.00)	0.068(0.89)	0.031(1.94)
Baroclinic (I)	0.059(1.00)	0.063(0.95)	0.032(1.85)
N.L./Diffusion (J)	0.057(1.00)	0.063(0.90)	0.031(1.84)
Fine Sediment*	0.049(1.00)	0.053(0.92)	0.025(1.95)
Coarse Sediment*	0.047(1.00)	0.050(0.94)	0.024(1.96)
N.L./Diffusion (I)	0.045(1.00)	0.048(0.93)	0.024(1.89)
Salinity	0.039(1.00)	0.043(0.92)	0.024(1.63)
Layer Vel.(v)	0.037(1.00)	0.039(0.97)	0.020(1.89)
Interpolation	0.027(1.00)	0.027(1.01)	0.017(1.61)
Layer Vel.(u)	0.024(1.00)	0.025(0.96)	0.012(1.97)
Integrated Vel. (U)	0.016(1.00)	0.016(1.00)	0.011(1.42)
Dimensionalize	0.014(1.00)	0.013(1.04)	0.008(1.80)
Integrate. Vel. (V)	0.013(1.00)	0.012(1.04)	0.007(1.80)
Layer Vel.(w)	0.013(1.00)	0.013(0.99)	0.007(1.88)
Bottom Shear Stress*	0.001(1.00)	0.001(0.99)	0.000(1.83)
Wave H/T*	0.000(1.00)	0.000(0.99)	0.000(1.91)
All Parallel Routines	1.829(1.00)	1.856(0.99)	0.981(1.87)
Total Runtime	1.841(1.00)	1.867(0.99)	0.997(1.85)

Table N.9: CPU times for the parallel, shared memory, CH3D procedures which are used in a hydrodynamic, salinity, sediment and nutrient simulation on the "nereus" computer. Times shown are per time step iteration of the model using the boundary-fitted "fine grid" (477x43) and and are given in seconds, n is the number of processors used and speedup is shown in parenthesis. Times for procedures marked with a "*" are included with other procedures are not thus not included in the totals.

Procedure	Serial	$n = 1$	$n = 2$
Main WQ	1.136(1.00)	1.133(1.00)	0.604(1.88)
Nutrient Transport*	0.310(1.00)	0.308(1.01)	0.165(1.88)
Main Sediment	0.119(1.00)	0.118(1.01)	0.073(1.63)
Turbulence	0.072(1.00)	0.071(1.02)	0.037(1.98)
Baroclinic (I)	0.051(1.00)	0.051(0.99)	0.027(1.88)
N.L./Diffusion (I)	0.044(1.00)	0.044(1.00)	0.024(1.85)
N.L./Diffusion (J)	0.044(1.00)	0.044(1.00)	0.023(1.97)
Baroclinic (J)	0.043(1.00)	0.042(1.00)	0.022(1.92)
Fine Sediment*	0.038(1.00)	0.038(1.01)	0.020(1.91)
Salinity	0.037(1.00)	0.036(1.02)	0.022(1.71)
Coarse Sediment*	0.037(1.00)	0.036(1.01)	0.019(1.91)
Interpolation	0.034(1.00)	0.033(1.02)	0.018(1.89)
Layer Vel.(v)	0.025(1.00)	0.025(0.99)	0.013(1.92)
Layer Vel.(u)	0.020(1.00)	0.021(0.99)	0.010(1.95)
Layer Vel.(w)	0.015(1.00)	0.015(0.99)	0.008(1.96)
Integrated Vel. (U)	0.015(1.00)	0.015(0.99)	0.010(1.50)
Integrate. Vel. (V)	0.011(1.00)	0.011(0.99)	0.006(1.90)
Dimensionalize	0.009(1.00)	0.009(1.00)	0.005(1.80)
Bottom Shear Stress*	0.001(1.00)	0.001(0.99)	0.000(1.96)
Wave H/T*	0.001(1.00)	0.001(0.98)	0.000(1.93)
All Parallel Routines	1.676(1.00)	1.670(1.00)	0.901(1.86)
Total Runtime	1.677(1.00)	1.672(1.00)	0.911(1.84)

Table N.10: CPU times for the parallel, shared memory, CH3D procedures which are used in a hydrodynamic, salinity, sediment and nutrient simulation on the "ocean" computer. Times shown are per time step iteration of the model using the boundary-fitted "coarse grid" (198x22) and and are given in seconds, n is the number of processors used and speedup is shown in parenthesis. Times for procedures marked with a "*" are included with other procedures are not thus not included in the totals.

Procedure	Serial	$n = 1$	$n = 2$	$n = 3$	$n = 4$
Main WQ	0.302(1.00)	0.317(0.95)	0.181(1.67)	0.126(2.40)	0.113(2.67)
Nutrient Transport*	0.084(1.00)	0.098(0.86)	0.054(1.57)	0.038(2.24)	0.034(2.47)
Main Sediment	0.033(1.00)	0.036(0.95)	0.021(1.54)	0.015(2.13)	0.014(2.33)
Turbulence	0.016(1.00)	0.019(0.87)	0.009(1.76)	0.006(2.67)	0.005(3.15)
Baroclinic (I)	0.014(1.00)	0.014(0.99)	0.008(1.76)	0.005(2.57)	0.005(2.91)
N.L./Diffusion (J)	0.014(1.00)	0.013(1.03)	0.007(1.87)	0.005(2.70)	0.004(3.34)
Fine Sediment*	0.012(1.00)	0.013(0.91)	0.007(1.71)	0.005(2.59)	0.004(2.93)
Coarse Sediment*	0.011(1.00)	0.012(0.92)	0.007(1.66)	0.005(2.50)	0.004(2.77)
Baroclinic (J)	0.011(1.00)	0.012(0.91)	0.006(1.71)	0.005(2.35)	0.004(2.60)
N.L./Diffusion (I)	0.011(1.00)	0.011(1.01)	0.006(1.68)	0.005(2.33)	0.004(2.72)
Salinity	0.010(1.00)	0.011(0.92)	0.007(1.37)	0.005(1.80)	0.005(1.93)
Layer Vel.(v)	0.009(1.00)	0.009(0.96)	0.005(1.69)	0.004(2.39)	0.003(2.97)
Interpolation	0.007(1.00)	0.007(1.00)	0.004(1.61)	0.003(2.25)	0.003(2.47)
Layer Vel.(u)	0.006(1.00)	0.006(0.95)	0.004(1.60)	0.002(2.34)	0.002(2.82)
Dimensionalize	0.004(1.00)	0.004(0.97)	0.002(1.70)	0.001(2.63)	0.001(2.87)
Integrated Vel. (U)	0.004(1.00)	0.004(1.02)	0.003(1.19)	0.003(1.40)	0.002(1.48)
Layer Vel.(w)	0.003(1.00)	0.003(0.97)	0.002(1.53)	0.001(2.31)	0.001(2.59)
Integrate. Vel. (V)	0.002(1.00)	0.002(0.94)	0.002(1.35)	0.001(1.80)	0.001(2.07)
Bottom Shear Stress*	0.000(1.00)	0.000(1.02)	0.000(1.79)	0.000(2.73)	0.000(3.08)
Wave H/T*	0.000(1.00)	0.000(1.04)	0.000(1.84)	0.000(2.89)	0.000(3.28)
All Parallel Routines	0.444(1.00)	0.466(0.95)	0.268(1.65)	0.188(2.36)	0.168(2.64)
Total Runtime	0.446(1.00)	0.469(0.95)	0.273(1.63)	0.194(2.30)	0.174(2.57)

Table N.11: CPU times for the parallel, shared memory, CH3D procedures which are used in a hydrodynamic, salinity, sediment and nutrient simulation on the "seiche" computer. Times shown are per time step iteration of the model using the boundary-fitted "coarse grid" (198x22) and and are given in seconds, n is the number of processors used and speedup is shown in parenthesis. Times for procedures marked with a "*" are included with other procedures are not thus not included in the totals.

Procedure	Serial	$n = 1$	$n = 2$
Main WQ	0.278(1.00)	0.290(0.96)	0.161(1.73)
Nutrient Transport*	0.081(1.00)	0.090(0.89)	0.047(1.70)
Main Sediment	0.031(1.00)	0.034(0.92)	0.019(1.63)
Turbulence	0.015(1.00)	0.017(0.88)	0.009(1.78)
Baroclinic (I)	0.013(1.00)	0.013(0.99)	0.007(1.80)
N.L./Diffusion (J)	0.012(1.00)	0.012(1.03)	0.006(1.91)
Fine Sediment*	0.012(1.00)	0.013(0.90)	0.006(1.80)
Coarse Sediment*	0.011(1.00)	0.012(0.92)	0.006(1.80)
Baroclinic (J)	0.010(1.00)	0.011(0.92)	0.006(1.78)
N.L./Diffusion (I)	0.010(1.00)	0.010(1.01)	0.006(1.70)
Salinity	0.010(1.00)	0.010(0.96)	0.007(1.49)
Layer Vel.(v)	0.008(1.00)	0.008(0.96)	0.004(1.77)
Interpolation	0.006(1.00)	0.006(1.00)	0.004(1.68)
Layer Vel.(u)	0.005(1.00)	0.006(0.96)	0.003(1.74)
Integrated Vel. (U)	0.004(1.00)	0.004(1.00)	0.003(1.30)
Dimensionalize	0.003(1.00)	0.003(0.98)	0.002(1.52)
Layer Vel.(w)	0.003(1.00)	0.003(0.99)	0.002(1.70)
Integrate. Vel. (V)	0.002(1.00)	0.002(0.95)	0.001(1.53)
Bottom Shear Stress*	0.000(1.00)	0.000(1.03)	0.000(1.82)
Wave H/T*	0.000(1.00)	0.000(1.04)	0.000(1.86)
All Parallel Routines	0.412(1.00)	0.431(0.96)	0.240(1.72)
Total Runtime	0.416(1.00)	0.434(0.96)	0.245(1.69)

Table N.12: CPU times for the parallel, shared memory, CH3D procedures which are used in a hydrodynamic, salinity, sediment and nutrient simulation on the "nereus" computer. Times shown are per time step iteration of the model using the boundary-fitted "coarse grid" (198x22) and and are given in seconds, n is the number of processors used and speedup is shown in parenthesis. Times for procedures marked with a "*" are included with other procedures are not thus not included in the totals.

Procedure	Serial	$n = 1$	$n = 2$
Main WQ	0.273(1.00)	0.272(1.00)	0.152(1.80)
Nutrient Transport*	0.071(1.00)	0.071(1.00)	0.040(1.80)
Main Sediment	0.027(1.00)	0.027(1.00)	0.016(1.67)
Turbulence	0.017(1.00)	0.017(1.00)	0.009(1.92)
Baroclinic (I)	0.012(1.00)	0.012(0.99)	0.006(1.84)
N.L./Diffusion (I)	0.010(1.00)	0.010(1.00)	0.005(1.79)
Fine Sediment*	0.009(1.00)	0.009(1.02)	0.005(1.83)
N.L./Diffusion (J)	0.009(1.00)	0.009(0.99)	0.005(1.95)
Salinity	0.009(1.00)	0.008(1.04)	0.005(1.61)
Baroclinic (J)	0.009(1.00)	0.009(1.00)	0.005(1.85)
Coarse Sediment*	0.009(1.00)	0.009(1.01)	0.005(1.85)
Interpolation	0.008(1.00)	0.008(1.01)	0.005(1.78)
Layer Vel.(v)	0.006(1.00)	0.006(0.99)	0.003(1.88)
Layer Vel.(u)	0.005(1.00)	0.005(0.98)	0.003(1.76)
Layer Vel.(w)	0.004(1.00)	0.004(1.00)	0.002(1.83)
Integrated Vel. (U)	0.003(1.00)	0.003(1.00)	0.002(1.45)
Dimensionalize	0.002(1.00)	0.002(1.01)	0.002(1.58)
Integrate. Vel. (V)	0.002(1.00)	0.002(1.00)	0.001(1.65)
Bottom Shear Stress*	0.000(1.00)	0.000(0.95)	0.000(1.72)
Wave H/T*	0.000(1.00)	0.000(1.01)	0.000(1.91)
All Parallel Routines	0.394(1.00)	0.393(1.00)	0.221(1.79)
Total Runtime	0.395(1.00)	0.394(1.00)	0.224(1.77)

APPENDIX O
FLOWCHARTS FOR THE PEM

The following flowcharts illustrate the order of the operations performed in the PEM model. Figures O.1 through O.3 illustrate the flow of the main program while Figures O.4 and O.5 illustrates only the part of the implicit model responsible for calculating hydrodynamics.
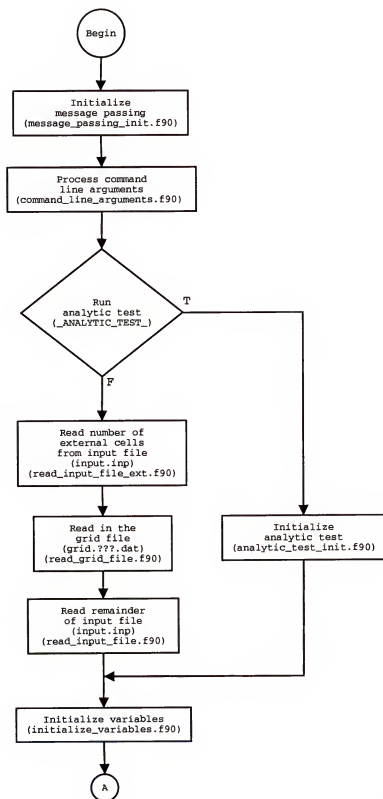
Figure O.1: Flowchart (Part 1/3) for implicit version of the Parallel Environmental Model.
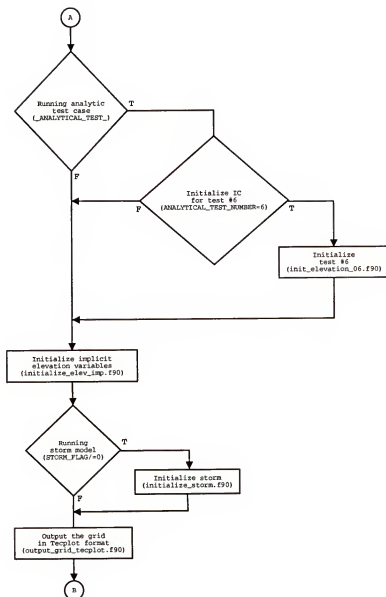
Figure O.2: Flowchart (Part 2/3) for implicit version of the Parallel Environmental Model.
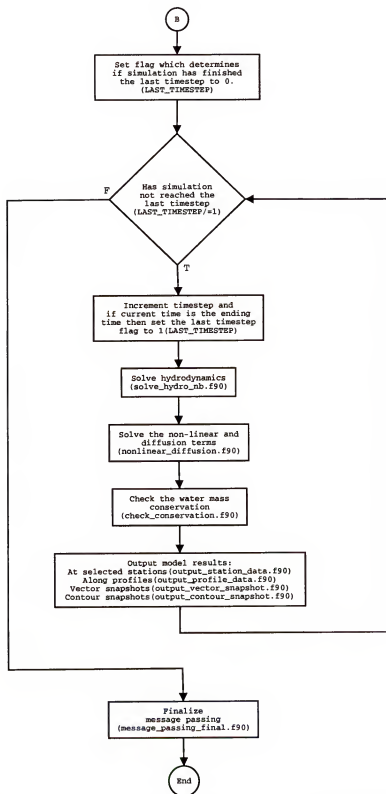
Figure O.3: Flowchart (Part 3/3) for implicit version of the Parallel Environmental Model.
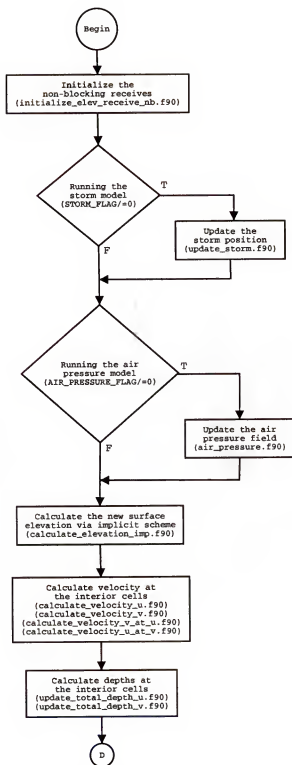
Figure O.4: Flowchart for the solution of the hydrodynamics of the implicit Parallel Environmental Model (Part 1/2).

```
                                    ┌─┐
                                    │D│
                                    └─┘
                          ┌────────────────────────┐
                          │ Complete the non-blocking│
                          │ receives of elevation   │
                          │[complete_elev_receive_nb.f90]│
                          └────────────────────────┘


                                   ╱╲
                                  ╱  ╲
                         ╱ Does the    ╲
                    F   ╱ local processor have╲   T
              ┌────────╱   an even rank        ╲────────┐
              │        ╲   (EVEN_RANK=1)        ╱        │
              │         ╲                      ╱         │
              │          ╲                    ╱          │
    ┌─────────────────┐   ╲╱              ┌─────────────────┐
    │Receive boundary velocities│          │Send boundary velocities│
    │from neighbors, SNEW │          │to neighbors, NSEW │
    │[receive_velocity_blocking.f90]│     │[send_velocity_blocking.f90]│
    └─────────────────┘                   └─────────────────┘

    ┌─────────────────┐                   ┌─────────────────┐
    │Send boundary velocities│            │Receive boundary velocities│
    │to neighbors, NSEW │                 │from neighbors, SNEW │
    │[send_velocity_blocking.f90]│        │[receive_velocity_blocking.f90]│
    └─────────────────┘                   └─────────────────┘
```

Calculate the velocities
along the boundaries
W[calculate_velocity_u.f90]
E[calculate_velocity_u.f90]
S[calculate_velocity_v.f90]
N[calculate_velocity_v.f90]
W[calculate_velocity_v_at_u.f90]
E[calculate_velocity_v_at_u.f90]
S[calculate_velocity_u_at_v.f90]
N[calculate_velocity_u_at_v.f90]

Update the total
depth along the
boundaries
W[update_total_depth_u.f90]
E[update_total_depth_u.f90]
S[update_total_depth_v.f90]
N[update_total_depth_v.f90]

Calculate velocities
along the open
boundaries
[calculate_velocity_ob.f90]

(Return)

Figure O.5: Flowchart for the solution of the hydrodynamics of the implicit Parallel Environmental Model (Part 2/2).

APPENDIX P
HURRICANE FLOYD WATER LEVEL COMPARISONS

Using the PEM, the following figures compare the simulated water level with the measured water level within the Indian River Lagoon during the passage of Hurricane Floyd. Figures P.1 through P.4 compare the water levels at the Banana River Station (FDEP #872-1789), Figures P.5 through P.8 compare the water levels at the Melbourne Causeway Station (FDEP #872-1843), and Figures P.9 through P.12 compare the water levels at the Sebastian Inlet Station (FDEP #872-2004). The filters used to process the water level (Doodson and Warburg, 1941; Panofsky and Brier, 1963) were the same as those used to filter the Sebastian Inlet water level used as tidal forcing dsecribed in Chapter 7.

Figure P.1: A comparison between simulated and measured water level at the Banana River Station (FDEP #872-1789) during the passage of Hurricane Floyd using the coarse grid. Simulated hurricanes were generated using two types of radius to maxium winds, a) a constant value of 20 n. mi. and b) the measured values recorded in the Vortex Data Messages.

a)

b)

Figure P.2: A comparison between the filtered simulated and measured water level at the Banana River Station (FDEP #872-1789) during the passage of Hurricane Floyd using the coarse grid. Simulated hurricanes were generated using two types of radius to maximum winds, a) a constant value of 20 n. mi. and b) the measured values recorded in the Vortex Data Messages.

Figure P.3: A comparison between simulated and measured water level at the Banana River Station (FDEP #872-1789) during the passage of Hurricane Floyd using the fine grid. Simulated hurricanes were generated using two types of radius to maximum winds, a) a constant value of 20 n. mi. and b) the measured values recorded in the Vortex Data Messages.

Figure P.4: A comparison between the filtered simulated and measured water level at the Banana River Station (FDEP #872-1789) during the passage of Hurricane Floyd using the fine grid. Simulated hurricanes were generated using two types of radius to maxium winds, a) a constant value of 20 n. mi. and b) the measured values recorded in the Vortex Data Messages.

Figure P.5: A comparison between simulated and measured water level at the Melbourne Causeway Station (FDEP #872-1843) during the passage of Hurricane Floyd using the coarse grid. Simulated hurricanes were generated using two types of radius to maxium winds, a) a constant value of 20 n. mi. and b) the measured values recorded in the Vortex Data Messages.

Figure P.6: A comparison between the filtered simulated and measured water level at the Melbourne Causeway Station (FDEP #872-1843) during the passage of Hurricane Floyd using the coarse grid. Simulated hurricanes were generated using two types of radius to maxium winds, a) a constant value of 20 n. mi. and b) the measured values recorded in the Vortex Data Messages.

Figure P.7: A comparison between simulated and measured water level at the Melbourne Causeway Station (FDEP #872-1843) during the passage of Hurricane Floyd. The water level was simulated using the fine grid. Simulated hurricanes were generated using two types of radius to maximum winds, a) a constant value of 20 n. mi. and b) the measured values recorded in the Vortex Data Messages.

Figure P.8: A comparison between the filtered simulated and measured water level at the Melbourne Causeway Station (FDEP #872-1843) during the passage of Hurricane Floyd using the fine grid. Simulated hurricanes were generated using two types of radius to maximum winds, a) a constant value of 20 n. mi. and b) the measured values recorded in the Vortex Data Messages.
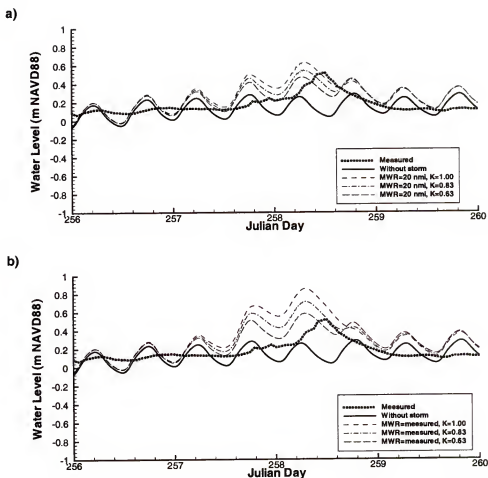
Figure P.9: A comparison between simulated and measured water level at the Sebastian Inlet Station (FDEP #872-2004) during the passage of Hurricane Floyd using the coarse grid. Simulated hurricanes were generated using two types of radius to maxium winds, a) a constant value of 20 n. mi. and b) the measured values recorded in the Vortex Data Messages.
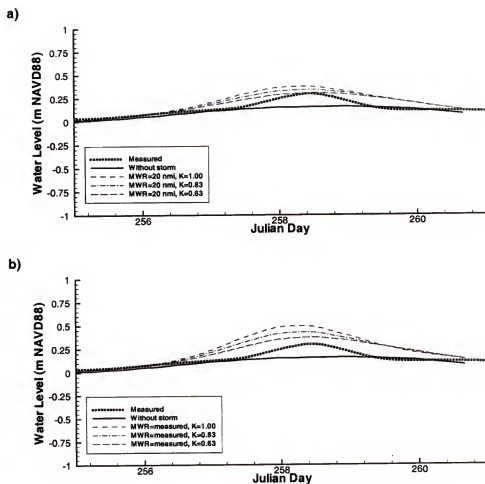
a)

b)

Figure P.10: A comparison between the filtered simulated and measured water level at the Sebastian Inlet Station (FDEP #872-2004) during the passage of Hurricane Floyd using the coarse grid. Simulated hurricanes were generated using two types of radius to maxium winds, a) a constant value of 20 n. mi. and b) the measured values recorded in the Vortex Data Messages.
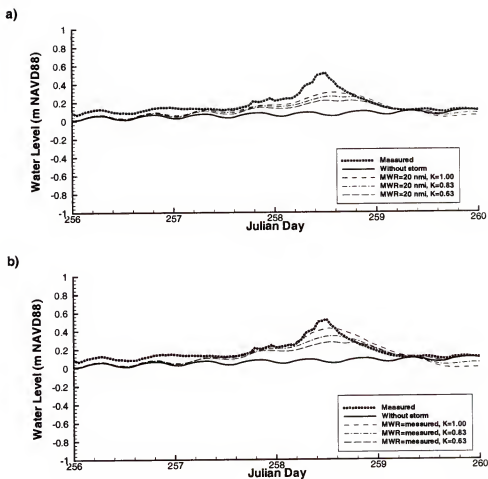
Figure P.11: A comparison between simulated and measured water level at the Sebastian Inlet Station (FDEP #872-2004) during the passage of Hurricane Floyd using the fine grid. Simulated hurricanes were generated using two types of radius to maximum winds, a) a constant value of 20 n. mi. and b) the measured values recorded in the Vortex Data Messages.
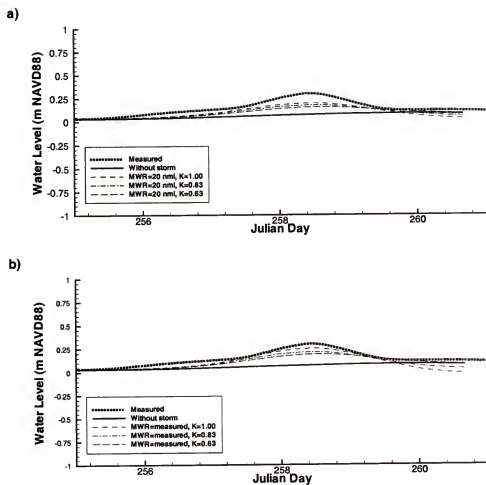
Figure P.12: A comparison between the filtered simulated and measured water level at the Sebastian Inlet Station (FDEP #872-2004) during the passage of Hurricane Floyd using the fine grid. Simulated hurricanes were generated using two types of radius to maxium winds, a) a constant value of 20 n. mi. and b) the measured values recorded in the Vortex Data Messages.
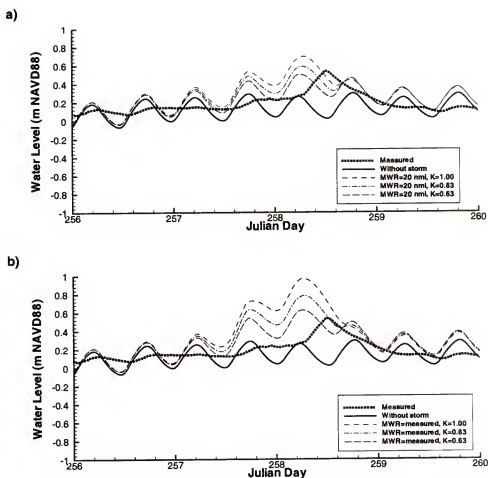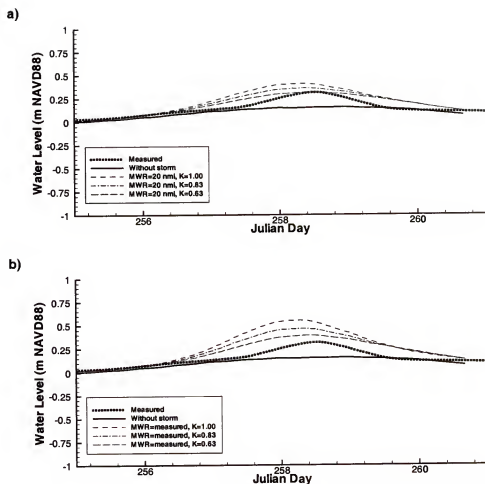
HURRICANE IRENE WATER LEVEL COMPARISONS

Using the PEM, the following figures compare the simulated water level with the measured water level within the Indian River Lagoon during the passage of Hurricane Irene. Figures Q.1 through Q.4 compare the water levels at the Banana River Station (FDEP #872-1789), Figures Q.5 through Q.8 compare the water levels at the Melbourne Causeway Station (FDEP #872-1843), and Figures Q.9 through Q.12 compare the water levels at the Sebastian Inlet Station (FDEP #872-2004). The filters used to process the water level (Doodson and Warburg, 1941; Panofsky and Brier, 1963) were the same as those used to filter the Sebastian Inlet water level used as tidal forcing dsecribed in Chapter 7.

Figure Q.1: A comparison between simulated and measured water level at the Banana River Station (FDEP #872-1789) during the passage of Hurricane Irene using the coarse grid. Simulated hurricanes were generated using two types of radius to maxium winds, a) a constant value of 20 n. mi. and b) the measured values recorded in the Vortex Data Messages.
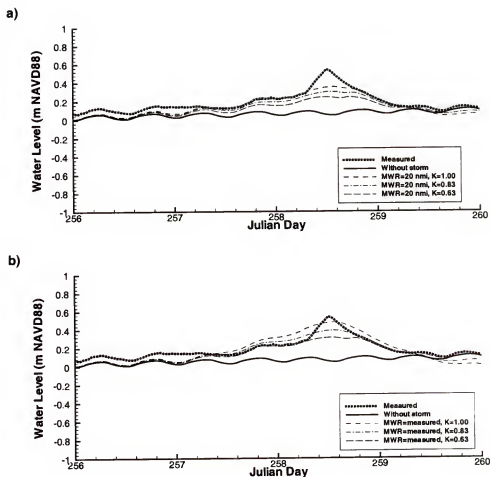
Figure Q.2: A comparison between the filtered simulated and measured water level at the Banana River Station (FDEP #872-1789) during the passage of Hurricane Irene using the coarse grid. Simulated hurricanes were generated using two types of radius to maxium winds, a) a constant value of 20 n. mi. and b) the measured values recorded in the Vortex Data Messages.
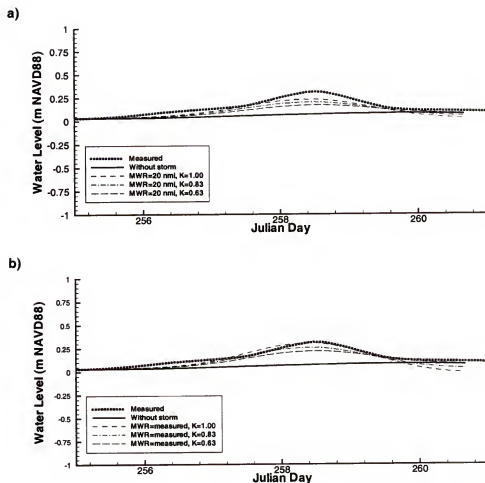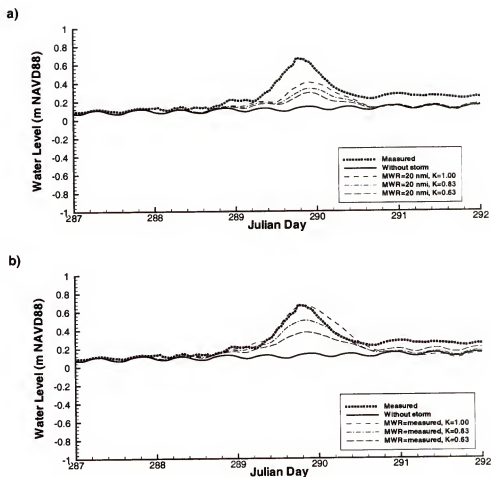
Figure Q.3: A comparison between simulated and measured water level at the Banana River Station (FDEP #872-1789) during the passage of Hurricane Irene using the fine grid. Simulated hurricanes were generated using two types of radius to maximum winds, a) a constant value of 20 n. mi. and b) the measured values recorded in the Vortex Data Messages.
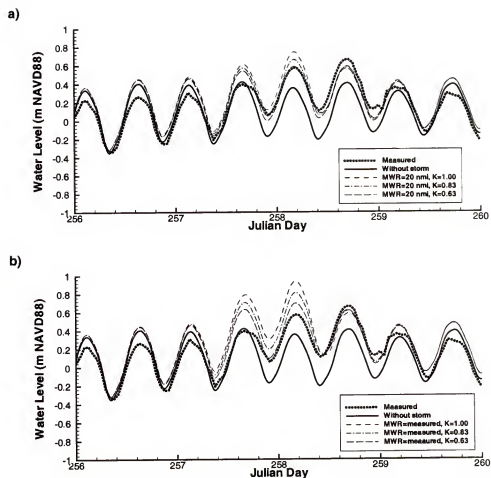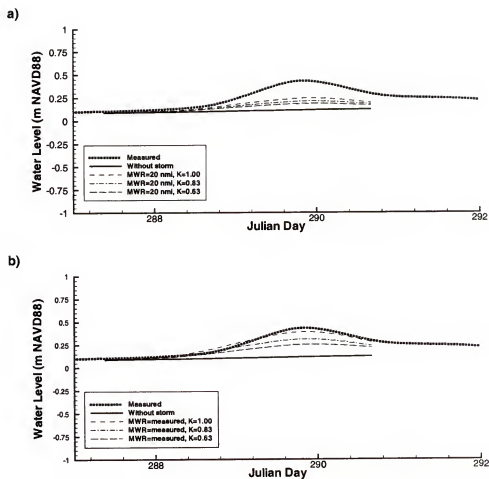
Figure Q.4: A comparison between the filtered simulated and measured water level at the Banana River Station (FDEP #872-1789) during the passage of Hurricane Irene using the fine grid. Simulated hurricanes were generated using two types of radius to maxium winds, a) a constant value of 20 n. mi. and b) the measured values recorded in the Vortex Data Messages.
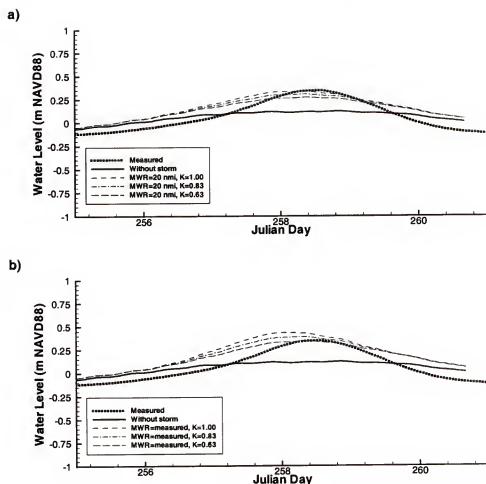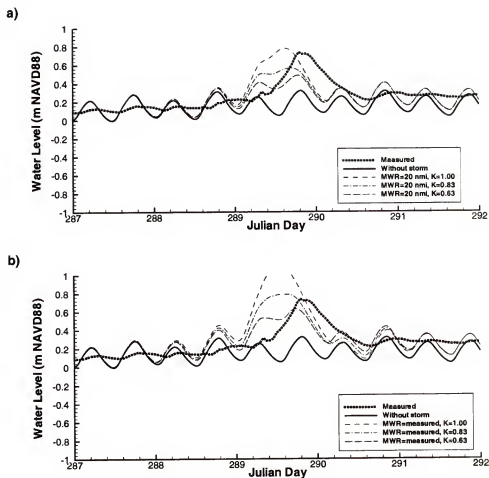
Figure Q.5: A comparison between simulated and measured water level at the Melbourne Causeway Station (FDEP #872-1843) during the passage of Hurricane Irene using the coarse grid. Simulated hurricanes were generated using two types of radius to maxium winds, a) a constant value of 20 n. mi. and b) the measured values recorded in the Vortex Data Messages.
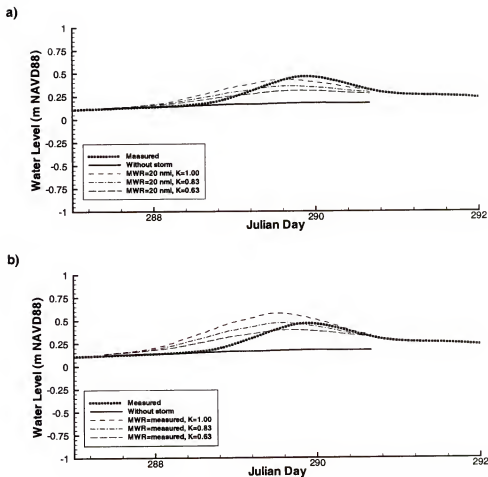
Figure Q.6: A comparison between the filtered simulated and measured water level at the Melbourne Causeway Station (FDEP #872-1843) during the passage of Hurricane Irene using the coarse grid. Simulated hurricanes were generated using two types of radius to maxium winds, a) a constant value of 20 n. mi. and b) the measured values recorded in the Vortex Data Messages.
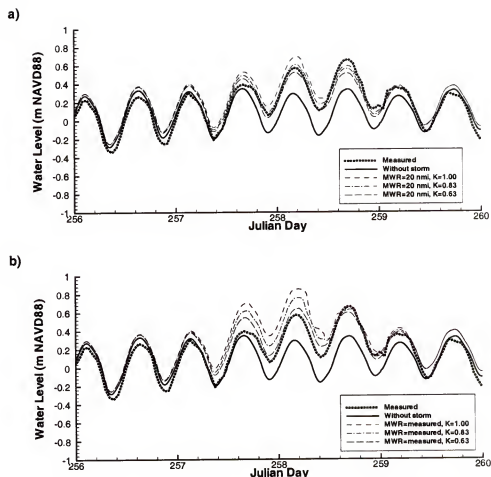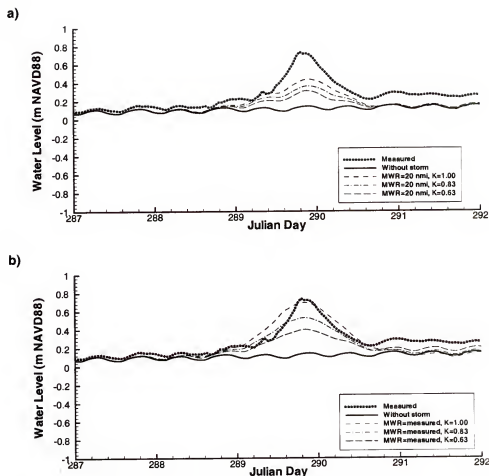
Figure Q.7: A comparison between simulated and measured water level at the Melbourne Causeway Station (FDEP #872-1843) during the passage of Hurricane Irene using the fine grid. Simulated hurricanes were generated using two types of radius to maximum winds, a) a constant value of 20 n. mi. and b) the measured values recorded in the Vortex Data Messages.
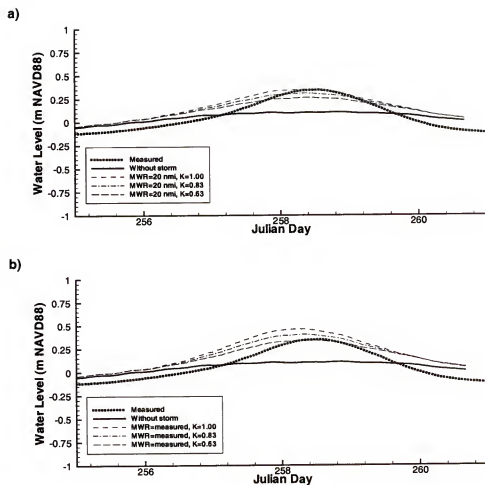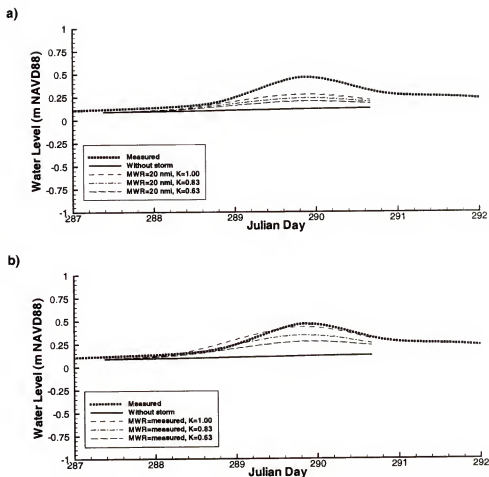
Figure Q.8: A comparison between the filtered simulated and measured water level at the Melbourne Causeway Station (FDEP #872-1843) during the passage of Hurricane Irene using the fine grid. Simulated hurricanes were generated using two types of radius to maxium winds, a) a constant value of 20 n. mi. and b) the measured values recorded in the Vortex Data Messages.
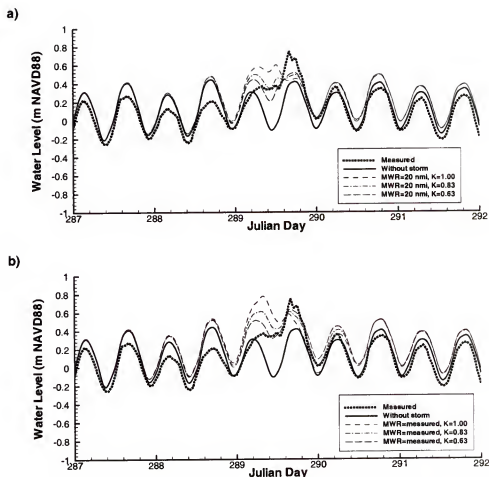
Figure Q.9: A comparison between simulated and measured water level at the Sebastian Inlet Station (FDEP #872-2004) during the passage of Hurricane Irene using the coarse grid. Simulated hurricanes were generated using two types of radius to maxium winds, a) a constant value of 20 n. mi. and b) the measured values recorded in the Vortex Data Messages.
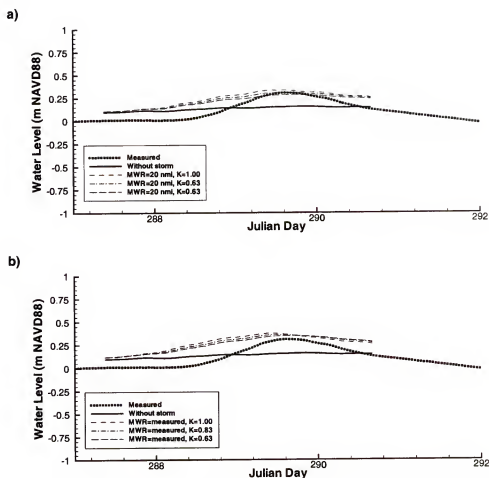
Figure Q.10: A comparison between the filtered simulated and measured water level at the Sebastian Inlet Station (FDEP #872-2004) during the passage of Hurricane Irene using the coarse grid. Simulated hurricanes were generated using two types of radius to maxium winds, a) a constant value of 20 n. mi. and b) the measured values recorded in the Vortex Data Messages.
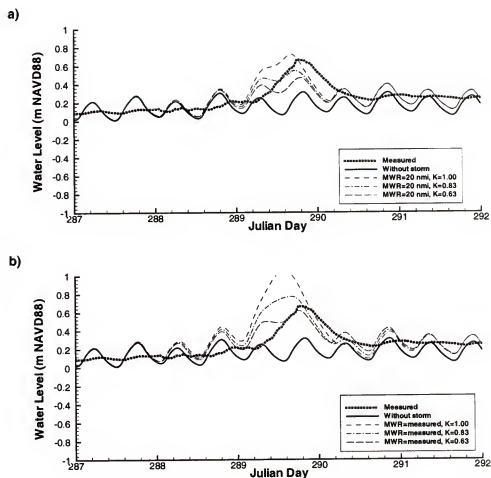
a)

b)
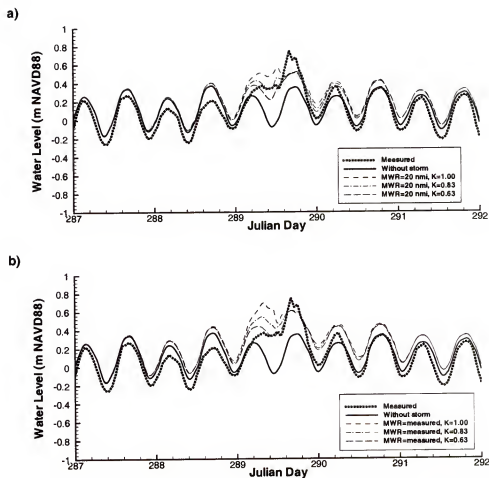
Figure Q.11: A comparison between simulated and measured water level at the Sebastian Inlet Station (FDEP #872-2004) during the passage of Hurricane Irene using the fine grid. Simulated hurricanes were generated using two types of radius to maximum winds, a) a constant value of 20 n. mi. and b) the measured values recorded in the Vortex Data Messages.
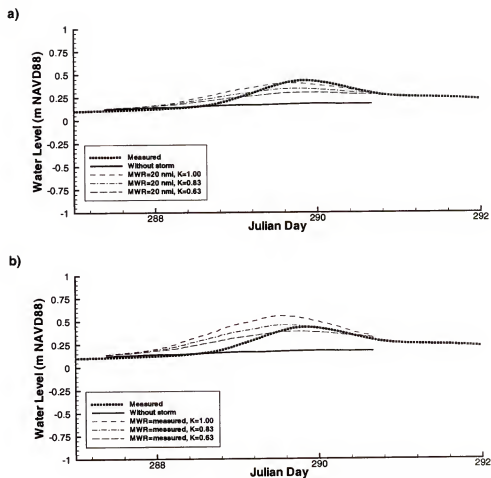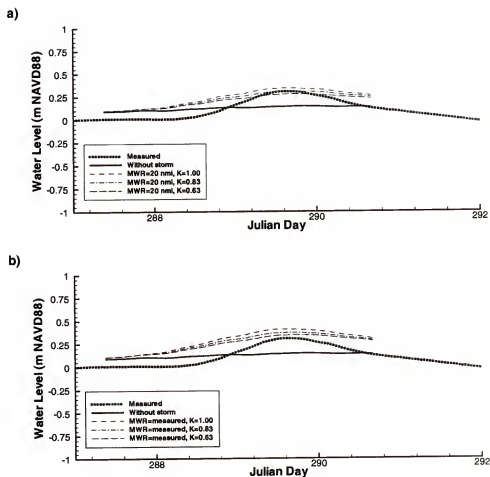
Figure Q.12: A comparison between the filtered simulated and measured water level at the Sebastian Inlet Station (FDEP #872-2004) during the passage of Hurricane Irene using the fine grid. Simulated hurricanes were generated using two types of radius to maxium winds, a) a constant value of 20 n. mi. and b) the measured values recorded in the Vortex Data Messages.

REFERENCES

Amdahl, G. (1967). Validity of the single processor approach to achieving large-scale computing capabilities. In *Proceedings of the American Federation of Information Processing Societies*, volume 30, pages 483–485, Washington, DC.

Anderson, E., Bai, Z., Bischof, C., Demmel, J., Dongarra, J., Croz, J. D., Greenbaum, A., Hammarling, S., McKenney, A., Ostrouchov, S., and Sorensen, D. (1995). *LAPACK Users' Guide*. SIAM, Philadelphia, Pennsylvania, 2nd edition.

Anderson, E., Bai, Z., Bischof, C., Demmel, J., Dongarra, J., Croz, J. D., Greenbaum, A., Hammarling, S., McKenney, A., and Sorensen, D. (1990). LAPACK: A portable linear algebra library for high-performance computers. In *Proceedings Supercomputing '90*, pages 2–11. IEEE Computer Society Press.

Anderson, E. C. and Dongarra, J. (1993). Performance of LAPACK: A portable library of numerical linear algebra routines. *Proceedings of the IEEE*, 81(8):1094–1102.

Avila, L. A. (1999). Hurricane Irene. Preliminary report, National Hurricane Center. http://www.nhc.noaa.gov/1999irene_text.html.

Barile, D., Panico, C. A., and Heath, G. (1987). Management plan and implementation strategy for the Indian River Lagoon systems. Technical report, Marine Resources Council, Florida Institute of Technology. Sea Grant Contract IRL IR-1.

Barnes, J. (1998). *Florida's Hurricane History*. The University of North Carolina Press, Chapel Hill, North Carolina.

Beck, R., Dean, S., O'Keefe, M., and Sawdey, A. (1995). A comparison of data-parallel and message-passing versions of the Miami Isopycnic Coordinate Ocean Model MICOM. *Parallel Computing*, 21(10):1695–1720.

Beguelin, A., Dongarra, J., Geist, A., Manchek, R., and Sunderam, V. (1991). *A Users' Guide To PVM (Parallel Virtual Machine)*. ORNL/TM–11826, Oak Ridge National Laboratory.

Behrens, J. (1996). An adaptive semi-Lagrangian advection scheme and its parallelization. *Monthly Weather Review*, 124:2386–2395.

Bleck, R. and Chassignet, E. (1994). Simulating the oceanic circulation with isopycnic-coordinate models. In Majumdar, S. K., Miller, E. W., Forbes, G. S., Schmalz, R. F., and Panah, A. A., editors, *The Oceans: Physical-Chemical Dynamics and Human Impact*, pages 17–39. The Pennsylvania Academy of Science.

Bleck, R., Rooth, C., Hu, D., and Smith, L. (1992). Salinity-driven thermocline transients in a wind- and thermohaline-forced isopycnic coordinate model of the North Atlantic. *Journal of Physical Oceanography*, 22:1486–1505.

Blumberg, A. F. and Mellor, G. L. (1980). A coastal ocean numerical model. In Sundermann, J. and Holz, K. P., editors, *Mathematical Modeling of Estuarine Physics, Proceedings of an International Symposium*, pages 203–219. Springer-Verlag, Berlin.

Boukas, L. A., Mimikou, N. T., Missirlis, N. M., Mellor, G. L., Lascaratos, A., and Korres, G. (2000). The parallelization of the Princeton Ocean Model. In Amestoy, P., Berger, P., Dayde, M., Duff, I., Fraysse, V., Giraud, L., and Ruiz, D., editors, *EURO-PAR'99: Parallel Processing. Lecture Notes in Computer Science Vol. 1685*, pages 1395–1402. Springer-Verlag; Berlin, Germany.

Carnahan, B., Luther, H. A., and Wilkes, J. O. (1969). *Applied Numerical Methods*. John Wiley, New York.

Carney, S., Heroux, M., and Li, G. (1993). A proposal for a sparse BLAS toolkit. Technical report, Cray Research Inc., Eagen, Minnesota.

Carrier, J. R. and Greenspan, H. P. (1958). Water waves of finite amplitude on a sloping beach. *Journal of Fluid Mechanics*, 4:97–109.

Casulli, V. and Cheng, R. T. (1992). Semi-implicit finite difference methods for three-dimensional shallow water flow. *International Journal for Numerical Methods in Fluids*, 15:629–648.

Chen, X. J. (1994). *Effects of hydrodynamics and sediment transport processes on nutrient dynamics in shallow lakes and estuaries*. PhD dissertation, University of Florida.

Chen, X. J. and Sheng, Y. P. (1996). Application of a coupled hydrodynamic-sediment-water quality model. In *Estuarine and Coastal Modeling*, volume 4, pages 325–339. American Society of Civil Engineers.

Choi, J.-K. (1992). *Three-dimensional curvilinear-grid modeling of baroclinic circulation and mixing in a partially mixed estuary*. PhD dissertation, University of Florida.

Davis, J. R. (1996). Hydrodynamic modeling in shallow water with wetting and drying. Master's thesis, University of Florida.

Davis, J. R. and Sheng, Y. P. (2000). High performance estuarine and coastal environmental modeling: The CH3D example. In *Estuarine and Coastal Modeling*, volume 6, pages 470–484. American Society of Civil Engineers.

Dean, R. G. and Chiu, T. Y. (1986). *Combined total storm tide frequency analysis for Brevard County, Florida*. Beaches and Shores Resource Center, Institute of Science and Public Affairs, Florida State University, Tallahassee, Florida.

Dongarra, J. J. (2001). Performance of various computers using standard linear equations software. Technical Report CS - 89 - 85, Computer Science Department, University of Tennessee, Knoxville, Tennessee. http://www.netlib.org/benchmark/.

Doodson, A. T. and Warburg, H. D. (1941). *Admiralty manual of tides*. Hydrographic Department, Her Majesty's Stationary Office, London.

Du, H. (2000). Use of three-dimensional hydrodynamic model and accurate advection scheme for modeling flushing dynamics and developing segmentation schemes in Indian River Lagoon. Master's thesis, University of Florida.

Eckart, C. (1958). Properties of water. Part II. The equation of state of water and sea water at low temperatures and pressures. *American Journal of Science*, 256(4):225–240.

Foreman, M. G. G. (1996). Manual for tidal heights analysis and prediction. Pacific Marine Science Report 77-10, Institute of Ocean Sciences, Patricia Bay, Victoria B. C. V8L 4B2 Canada. http://www.ios.bc.ca/ios/iap/pages/tidpack.htm.

Garratt, J. R. (1977). Review of drag coefficients over oceans and continents. *Monthly Weather Review*, 105:915–929.

Godin, G. (1972). *The analysis of tides.* University of Toronto Press, Toronto.

Godin, G. and Taylor, J. (1973). A simple method for the prediction of the time and height of high and low water. *International Hydrographical Review*, L(2).

Gropp, W. and Lusk, E. (2001). *User's Guide for mpich, A Portable Implementation of MPI: Version 1.2.2.* Mathematical and Computer Science Division, Argonne National Laboratory, University of Chicago. ftp://ftp.mcs.anl.gov/pub/mpi/userguide.ps.

Groves, G. W. (1955). Numerical filters for discrimination against tidal periodicities. *Transactions, American Geophysical Union*, 36(6):1073–1084.

Harleman, D. R. F. (1966). *Estuary and Coastline Hydrodynamics*, chapter Tidal Dynamics in Estuaries, Part II: Real Estuaries, pages 522–545. McGraw-Hill Book Company, Incorporated.

Holland, G. J. (1980). An analytic model of the wind and pressure profiles in hurricanes. *Monthly Weather Review*, 108(8):1212–1218.

Howell, G. L. (1997). A comprehensive field study of tidal inlet processes at Ponce de Leon Inlet, Florida. In Edge, B. L., editor, *Coastal Engineering 1996: Proceedings of the 25th International Conference*, volume 3, pages 3323–3336. American Society of Civil Engineers.

Hsu, S. A. (1995). A proper wind-stress drag coefficient formulation for computer modeling of seas and coastal regions. In Brebbia, C. A., Traversoni, L., and Wrobel, L. C., editors, *Computer Modeling of Seas and Coastal Regions II*, pages 413–420. Computational Mechanics Publications.

Jarrell, J. D., Hebert, P. J., and Mayfield, M. (1992). Hurricane experience levels of coastal county populations from texas to maine. Technical Memo NWS NHC 46, NOAA, Coral Gables, Florida.

Jarvinen, B. R., Neumann, C. J., and Davis, M. A. S. (1984). A tropical cyclone data tape for the North Atlantic Basin, 1886-1983: Contents, limitations, and uses. Technical Memorandum NWS NHC 22, National Oceanic and Atmospheric Administration, Washington, D. C.

King Jr., D. B., Smith, J. M., Militello, A., and Stauble, D. K. (1999). Ponce de Leon Inlet, Florida, Site investigation; Report 1, Selected portions of long-term measurements, 1995-1997. Technical Report CHL-99-1, U. S. Army Engineer Waterways Experiment Station, Vicksburg, MS 39180-6199.

Lakshmivarahan, S. and Dhall, S. K. (1990). *Analysis and Design of Parallel Algorithms: Arithmetic and Matrix Problems*. McGraw-Hill Publishing Company.

Landsea, C. W. (2000). FAQ : Hurricanes, typhoons and tropical cyclones. http://www.aoml.noaa.gov/hrd/tcfaq/tcfaqHED.html. Version 2.10.

Lee, H. K. and Sheng, Y. P. (1993). Modeling the effects of vegetation and thermal stratification on lake circulation. Technical report, Coastal and Oceanographic Engineering Department, University of Florida, Gainesville, Florida.

Lee, J. (2000). A three-dimensional conservative eulerian-lagrangian model for coastal and estuarine circulation. Master's thesis, University of Florida.

Lewellen, W. S. (1977). Use of invariant modeling. In Frost, W., editor, *Handbook of Turbulence*, volume 1. Plenum Publishing Corporation, New York.

Liu, Y. (1988). A two-dimensional finite-difference model for moving boundary hydrodynamic problems. Master's thesis, University of Florida.

Luo, Y. (1993). Three-dimensional moving-boundary hydrodynamic model. Engineer's thesis, University of Florida.

Luong, P., Breshears, C. P., and Gabb, H. A. (2000). Execution and load-balance improvements in the CH3D hydrodynamic simulation code. Technical Report 00-07, U.S. Army Engineer Research and Development Center, Major Shared Resource Center, Information Technology Laboratory, Vicksburg, Mississippi. http://www.wes.hpc.mil/pet/tech_reports/reports/report_index_bd.htm.

Mann, D. W. and Mehta, A. J. (1993). Constricting influence of tidal inlets on the measurement of sea level rise in bays. *Journal of Coastal Research*, SI(18):185–194.

Mehta, A. J. (1990). Significance of bay superelevation in measurement of sea level change. *Journal of Coastal Research*, 6(4):801–813.

Message Passing Interface Forum (1994). MPI: A message-passing interface standard. Technical Report MCS-P342-1193, Computer Science Department, University of Tennessee.

Militello, A. and Zarillo, G. A. (2000). Tidal motion in a complex inlet and bay system, Ponce de Leon Inlet, Florida. *Journal of Coastal Research*, 16(3):840–852.

Moustafa, M. Z. and Hamrick, J. M. (1994). Modeling circulation and salinity transport in the Indian River Lagoon. In *Estuarine and Coastal Modeling*, volume 3, pages 381–395. American Society of Civil Engineers.

Munk, W. H. and Anderson, E. R. (1948). Notes on a theory of the thermocline. *Journal of Marine Research*, 7(3):276–295.

Oliveira, A., Fortunato, A. B., and Baptista, A. M. (2000). Mass balance in Eulerian-Lagrangian transport simulations in estuaries. *Journal of Hydraulic Engineering*, 126(8):605–614.

Panofsky, H. A. and Brier, G. W. (1963). *Some applications of statistics to mete-orology.* Penn State University, University Park.

Pasch, R. J., Kimberlain, T. B., and Stewart, S. R. (1999). Hurricane Floyd. Preliminary report, National Hurricane Center. http://www.nhc.noaa.gov/1999floyd_text.html.

Peene, S. and Sheng, Y. P. (1995). Circulation and transport within a system of shallow, interconnected barrier island lagoons. Technical report, Coastal and Oceanographic Engineering Department, University of Florida, Gainesville, Florida.

Peene, S., Sheng, Y. P., and Houston, S. (1993). Modeling tidal and wind driven circulation in Sarasota and Tampa Bay. In *Estuarine and Coastal Modeling*, volume 2, pages 357–369. American Society of Civil Engineers.

Phillips, N. A. (1957). A coordinate system having some special advantages for numerical forecasting. *Journal of Meteorology*, 14(2):184–185.

Pielke, R. A. (1990). *The Hurricane.* Routledge, New York.

Rahman, M. (1983). Analytical solutions for tidal propagation in a rectangular basin. *Advances in Water Resources*, 6:44–53.

Ridge, D., Becker, D., Merkey, P., and Sterling, T. (1997). Beowulf: Harnessing the power of parallelism in a Pile-of-PCs. In *Aerospace Applications Conference Proceedings*, volume 2, pages 79–91. IEEE.

Shadid, J. N. and Tuminaro, R. S. (1992). Sparse iterative algorithm software for large-scale MIMD machines: An initial discussion and implementation. *Concurrency, Practice and Experience*, 4(6):481–489.

Sheng, Y. P. (1982). Hydraulic applications of a second-order closure model of turbulent transport. In Smith, P., editor, *Proceedings of the American Society of Civil Engineers Hydraulic Division Specialty Conference on Applying Research to Hydraulic Practice*, Jackson, Mississippi. American Society of Civil Engineers.

Sheng, Y. P. (1983). Mathematical modeling of three-dimensional coastal currents and sediment dispersion: Model development and application. Technical Report CERC-83-2, Aeronautical Research Associates of Princeton, Princeton, New Jersey.

Sheng, Y. P. (1984). A turbulent transport model of coastal processes. In *Proceedings of the 19th International Conference on Coastal Engineering*, pages 2380–2396. American Society of Civil Engineers.

Sheng, Y. P. (1985). A new one-dimensional ocean current model. Technical Report 556, Aeronautical Research Associates of Princeton, Princeton, New Jersey.

Sheng, Y. P. (1986). Modeling coastal and estuarine processes using boundary-fitted grids. In Wang, Shen, and Ding, editors, *River Sedimentation: Proceedings Third International Symposium on River Sedimentation*, pages 1426–1442, University of Mississippi, University, Mississippi.

Sheng, Y. P. (1987). On modeling three-dimensional estuarine and marine hydrodynamics. In Nihoul and Jamart, editors, *Three-dimensional models of marine and estuarine dynamics*, pages 35–54. Elsevier, Amsterdam.

Sheng, Y. P. (1990). Evolution of a three-dimensional curvilinear-grid hydrodynamic model for estuaries, lakes and coastal waters: CH3D. In *Estuarine and Coastal Modeling*, pages 40–49. American Society of Civil Engineers, New York.

Sheng, Y. P. (1994). Modeling hydrodynamics and water quality dynamics in shallow waters. In *Proceedings of the First International Symposium on Ecology and Engineering*, Taman Negara, Malaysia. University of Western Australia and Malaysian Technical University.

Sheng, Y. P. (1995). On modeling circulation in Florida Bay. In *American Geophysical Union Conference on Circulation in Intra-American Seas*, Puerto Rico. American Geophysical Union.

Sheng, Y. P. (1997). Development of a preliminary hydrodynamics and water quality model of the Indian River Lagoon. Final Report to the St. Johns River Water Management District, Coastal and Oceanographic Engineering Department, University of Florida, Gainesville, Florida.

Sheng, Y. P. (2000). A framework for integrated modeling of coupled hydrodynamic-sedimentary-ecological processes. In *Estuarine and Coastal Modeling*, volume 6, pages 350–362. American Society of Civil Engineers.

Sheng, Y. P., Ahn, K. M., and Choi, J.-K. (1990a). Wind-wave hindcasting and estimation of bottom shear stress in Lake Okeechobee. Technical Report UFL/COEL-93/021, Coastal and Oceanographic Engineering Department, University of Florida, Gainesville, Florida.

Sheng, Y. P., Barry, K., Davis, J. R., Liu, Y.-F., Zhong, J., and Ge, R. (1996a). A preliminary hydrodynamics and water quality model of Indian River Lagoon. Final Report of Project #94W241, Coastal and Oceanographic Engineering Department, University of Florida, Gainesville, Florida.

Sheng, Y. P., Chen, X. J., Schofield, S., and Yassuda, E. (1993a). Hydrodynamics, sediment and phosphorus dynamics in Lake Okeechobee during an episodic event. Technical report, Coastal and Oceanographic Engineering Department, University of Florida, Gainesville, Florida.

Sheng, Y. P., Chen, X. J., Yassuda, E., Reddy, K. R., and Fisher, M. M. (1993b). Quantifying sediment resuspension flux of nutrients and contaminants in estuaries due to episodic events. Technical report, Coastal and Oceanographic Engineering Department, University of Florida, Gainesville, Florida.

Sheng, Y. P. and Chiu, S. S. (1986). Tropical cyclone generated currents. In *Proceedings of the 20th International Conference on Coastal Engineering*, pages 737–751. American Society of Civil Engineers.

Sheng, Y. P., Choi, J.-K., and Kuo, A. Y. (1990b). Three-dimensional numerical modeling of tidal circulation and salinity transport in James River Estuary. In Spaulding, M. L., editor, *Estuarine and Coastal Modeling*, pages 209–218. American Society of Civil Engineers, New York.

Sheng, Y. P., Cook, V., Peene, S., Eliason, D., Schofield, S., Ahn, K. M., and Wang, P. F. (1990c). A field and modeling study of fine sediment transport in shallow waters. In Spaulding, M. L., editor, *Estuarine and Coastal Modeling*, pages 113–122. American Society of Civil Engineers, New York.

Sheng, Y. P. and Davis, J. R. (1999). Measure nutrient concentration/ssc/salinity at ocean boundary, task 5.4 of irl-plr model development. Interim report, Coastal and Oceanographic Engineering Department, University of Florida.

Sheng, Y. P., Du, H., Kornick, A., Davis, J. R., Sun, D., Qiu, C., Melanson, J., and Christian, D. (1998a). Resuspension of sediments and nutrients in the Indian River Lagoon. Interim report on first episodic event, Civil and Coastal Engineering Department, University of Florida.

Sheng, Y. P. and Lee, H. K. (1991). Computation of phosphorus flux between the vegetation area and the open water in Lake Okeechobee. Technical Report UFL/COEL-91/022, Coastal and Oceanographic Engineering Department, University of Florida, Gainesville, Florida.

Sheng, Y. P., Lee, H. K., and Demas, C. E. (1994). Simulation of flushing in Indian River Lagoon using 1-d and 3-d models. In *Estuarine and Coastal Modeling*, volume 3, pages 366–380. American Society of Civil Engineers.

Sheng, Y. P., Melanson, J., Sun, D., Qiu, C., Davis, J., Du, H., Christian, D., and Kornick, A. (1999). Enhanced water quality sampling. Interim report to the st. johns river water management district, Civil and Coastal Engineering Department, University of Florida.

Sheng, Y. P. and Meng, X. (1993). A three-dimensional hydrodynamic numerical model of wind-driven circulation in Lake Apopka (LAHM3D), Volume V: Theory, model documentation, user's manual and model simulation. Technical Report UFL/COEL-93/006, Coastal and Oceanographic Engineering Department, University of Florida, Gainesville, Florida.

Sheng, Y. P. and Peene, S. (1994). A field and modeling study of residual circulation in Sarasota and Tampa Bay. In *Estuarine and Coastal Modeling*, volume 3, pages 641–655. American Society of Civil Engineers.

Sheng, Y. P., Peene, S., and Liu, Y. M. (1990d). Numerical modeling of tidal hydrodynamics and salinity transport in the Indian River Lagoon. *Florida Scientist*, 53(3):147–168.

Sheng, Y. P., Sun, D., Henderson, M., and Melanson, J. (1998b). Sediment map of the Indian River Lagoon. Interim report to the st. johns river water management district, Civil and Coastal Engineering Department, University of Florida.

Sheng, Y. P. and Villaret, C. (1989). Modeling the effect of suspended sediment stratification on bottom exchange processes. *Journal of Geophysical Research - Oceans*, 94(C10):14229–14444.

Sheng, Y. P. and Yassuda, E. (1994). Application of a 3-d hydrodynamic model to Tampa Bay. Technical report, Coastal and Oceanographic Engineering Department, University of Florida, Gainesville, Florida.

Sheng, Y. P., Yassuda, E. A., and Yang, C. (1995). Modeling the impact of nutrient load reduction on water quality and seagrass in Roberts Bay and Little Sarasota Bay. Technical report, Coastal and Oceanographic Engineering Department, University of Florida, Gainesville, Florida.

Sheng, Y. P., Yassuda, E. A., and Yang, C. (1996b). Modeling the effect of reduced nutrient loading on water quality in Roberts Bay. In *Estuarine and Coastal Modeling*, volume 4, pages 644–658. American Society of Civil Engineers.

Sigua, G. C., Steward, J. S., and Tweedale, W. A. (1996). Indian river lagoon water quality monitoring network proposed modifications. Memorandum 12, Department of Water Resources, St. Johns River Water Management District.

Simpson, R. H. and Riehl, H. (1981). *The Hurricane and Its Impact.* Louisiana State University Press, Baton Rouge, Louisiana.

SJRWMD, Department of Environmental Protection, Central Florida District, Brevard County Natural Resources Management Division, Indian River County Public Health Unit, and Volusia County Environmental Management Department (1993). *Indian River Lagoon Water Quality Monitoring Network Compiled Quality Assurance/Quality Control Manual for the Indian River Lagoon Surface Water Quality Monitoring Network.* Indian River Lagoon National Estuary Program, Melbourne, Florida.

Smith, N. P. (1987). An introduction to the tides of Florida's Indian River Lagoon. I. Water levels. *Florida Scientist*, 50(1):49–61.

Smith, S. D. and Banke, E. G. (1975). Variation of sea-surface drag coefficient with wind speed. *Quarterly Journal of the Royal Meteorological Society*, 101(429):665–673.

Sokolnikoff, I. S. (1960). *Tensor Analysis.* John Wiley and Sons, Incorporated, New York.

Staniforth, A. and Cote, J. (1991). Semi-Lagrangian integration schemes for atmospheric models-A review. *Monthly Weather Review*, 119(9):2206–2223.

Sterling, T., Savarese, D., Becker, D. J., and Dorband, J. E. (1995). BEOWULF: A parallel workstation for scientific computation. In *Proceedings of the International Conference on Parallel Processing*, volume 1, pages 11–14.

Steward, J. S. and Higman, J. (1991). *Master water quality monitoring network quality assurance/quality control manual for the Indian River Lagoon surface water quality monitoring network project 1988-1990.* SJRWMD, Palatka, Florida.

Sucsy, P. (2000). Personal communication. Discussion of methods used by the SJRWMD to recreate missing water level data.

Sucsy, P. and Morris, F. W. (1998). Proposed discharge limits for Turkey Creek, Brevard County, for maintaining a desirable salinity regime in the Indian River Lagoon. Technical Memorandum 26, Department of Water Resources, St. Johns River Water Management District, Palatka, Florida.

Sucsy, P. V., Morris, F. W., Bergman, M. J., and Donnangelo, L. J. (1998). A 3-d model of Florida's Sebastian River Estuary. In *Estuarine and Coastal Modeling*, volume 5, pages 59–74. American Society of Civil Engineers.

Sun, D. (2001). *Modeling suspended sediment transport under combined wave current interactions in Indian River Lagoon.* PhD dissertation, University of Florida.

Thompson, E. F. and Cardone, V. J. (1996). Practical modeling of hurricane surface wind fields. *Journal of Waterway, Port, Coastal and Ocean Engineering*, 122(4):195–205.

Thompson, J. E. (1982). *Numerical Grid Generation*, chapter General Curvilinear Coordinate systems, pages 1–30. Elsevier, Amsterdam.

Thompson, J. F. (1985). Development of an adaptive boundary-fitted coordinate code for use in coastal and estuarine areas. Miscellaneous paper HL-85-5, U. S. Army Engineer Waterways Experiment Station, Vicksburg, Mississippi.

Thompson, J. F., Warsi, Z. U. A., and Mastin, C. W. (1985). *Numerical Grid Generation: Foundations and applications.* Elsevier Science Publishing Company, Inc. http://WWW.ERC.MsState.Edu:80/education/gridbook/.

Tuminaro, R. S., Heroux, M., Hutchinson, S. A., and Shadid, J. N. (1999). Official Aztec User's Guide - Version 2.1. Technical Report SAND99-8801J, Sandia National Laboratories, Albuquerque, New Mexico 87185. http://www.cs.sandia.gov/CRF/aztec1.html.

Van Dorn, W. (1953). Wind stress on an artificial pond. *Journal of Marine Research*, 12(3):249–276.

Vickery, P. J., Skerlj, P. F., Steckley, A. C., and Twisdale, L. A. (2000). Hurricane wind field model for use in hurricane simulations. *Journal of Structural Engineering*, 126(10):1203–1221.

Vickery, P. J. and Twisdale, L. A. (1995). Prediction of hurricane wind speeds in the United States. *Journal of Structural Engineering*, 121(11):1691–1699.

Wang, H. V. and Kim, S.-C. (2000). Simulation of tunnel island and bridge piling effects in a tidal estuary. In *Estuarine and Coastal Modeling*, volume 6, pages 250–269. American Society of Civil Engineers.

Wilson, B. W. (1957). Hurricane wave statistics for the Gulf of Mexico. Technical Memorandum 98, Department of the Army Corps of Engineers.

Yassuda, E. and Sheng, Y. P. (1998). Modeling dissolved oxygen dynamics of Tampa Bay during summer of 1991. In *Estuarine and Coastal Modeling*, volume 5, pages 35–58. American Society of Civil Engineers.

Yassuda, E. A. (1996). *Integrated Modeling of the Tampa Bay Estuarine System.* PhD dissertation, University of Florida.

Zhu, J., Johnson, B., Bangalore, P., Huddleston, D., and Skjellum, T. (1998a). On the parallelization of CH3D. Technical Report 98-07, U.S. Army Corps of Engineers Waterways Experiment Station, Major Shared Resource Center, Information Technology Laboratory, Vicksburg, Mississippi. http://www.wes.hpc.mil/pet/tech_reports/reports/report_index_bd.htm.

Zhu, J., Johnson, B., Bangalore, P., Huddleston, D., and Skjellum, T. (1998b). On the parallelization of CH3D. In Abt, S. R., Young-Pezeshk, J., and Watson, C. C., editors, *Proceedings of the International Water Resources Engineering Conference*, volume 2, pages 1108–1113. American Society of Civil Engineers.
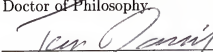
BIOGRAPHICAL SKETCH

Justin Ross Davis was born in San Francisco, California, on the 28th of June, 1972. He moved to Virginia Beach, VA, in 1977, where he continued to live through graduation from Cox High School in the spring of 1990. In the fall of 1990, he began college at the University of Virginia. Four years later, he obtained a Bachelor of Science in Aerospace Engineering from the Mechanical, Aerospace and Nuclear Engineering Department. The following summer he left for Gainesville, Florida, and began his graduate studies in coastal and oceanographic engineering at the University of Florida. He earned a master's in 1996 and a doctorate in 2001.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.
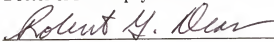
Y. Peter Sheng, Chairman
Professor of Civil and Coastal
Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.
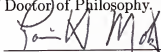
Timothy A. Davis
Associate Professor of Computer and
Information Science and Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Robert G. Dean
Graduate Research Professor of
Civil and Coastal Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Louis H. Motz
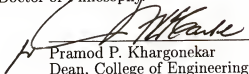Associate Professor of Civil and
Coastal Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Robert J. Thieke
Assistant Professor of Civil and
Coastal Engineering

This dissertation was submitted to the Graduate Faculty of the College of Engineering and to the Graduate School and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

December 2001

Pramod P. Khargonekar
Dean, College of Engineering

Winfred M. Phillips
Dean, Graduate School